



**Adriano Micael Jesus  
Gama**

**Domótica - Sistema Inteligente de Controlo e  
Monitorização**





**Adriano Micael Jesus  
Gama**

**Domótica - Sistema Inteligente de Controlo e  
Monotorização**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob orientação científica do Prof. Doutor José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.



**O júri / The jury**

Presidente / President

**Prof. Doutor Pedro Nicolau Faria da Fonseca**

Professor Auxiliar da Universidade de Aveiro

Vogais / Committee

**Prof. Doutor Arnaldo Silva Rodrigues de Oliveira**

Professor Auxiliar da Universidade de Aveiro (arguente principal)

**Prof. Doutor José Paulo Oliveira Santos**

Professor Auxiliar da Universidade de Aveiro (orientador)



**Agradecimentos /  
Acknowledgements**

Em primeiro lugar quero agradecer aos meus pais por me terem concedido esta oportunidade e pelo apoio, confiança e carinho incondicional que sempre me transmitiram.

Agradeço também aos amigos que me acompanharam neste percurso académico e aos amigos com os quais tive a oportunidade de vivenciar momentos marcantes.

Por fim quero agradecer ao Professor Doutor José Paulo Santos pela orientação, apoio e conhecimentos transmitidos.





**Palavras-chave**

Domótica; Sistema de monitorização e controlo; Movicon; MODBUS; Protocolo RS-485; Comunicação radiofrequência; Microcontroladores; Serviços WEB; Aconselhamento energético.

**Resumo**

A domótica é uma tecnologia que visa a gestão dos recursos habitacionais. Tem inúmeras aplicações práticas que vão desde o controlo de iluminação e climatização até à deteção de intrusão, tendo como principais vantagens as componentes de segurança, o conforto com que se executam os comandos e a racionalização de energia.

Esta dissertação propõe o desenvolvimento de um sistema de domótica para monitorização e controlo, de forma integrada e em tempo real, das várias tecnologias existentes num ambiente residencial.

O trabalho desenvolvido consistiu na criação de uma rede de domótica que combinava um meio cablado (par entrançado) com um meio sem fios (radiofrequência). Com recurso ao protocolo da camada de aplicação, MODBUS, aplicaram-se, na rede, módulos microcontrolados capazes de desempenhar funções de monitorização e atuação. Adotou-se uma arquitetura centralizada com recurso ao *software* SCADA Movicon, que possibilitava o acesso remoto ao sistema, a prestação de serviços WEB e a integração de dispositivos de diferentes fabricantes, pela sua compatibilidade com diversos protocolos de comunicação. A nível gráfico, a aplicação desenvolvida baseava-se na informação amostrada e nas bases de dados criadas para apresentar o estado atual das propriedades dos dispositivos e gerar animações conforme a interação do utilizador com as funcionalidades do sistema.



**Keywords**

Home Automation; Monitoring and control system; Movicon; MODBUS; RS-485 protocol; Radio frequency communication; Microcontrollers; WEB services; Energetic counseling.

**Abstract**

Home automation is a technology targeting the management of the habitational resources. It has numerous practical applications ranging from lighting and HVAC control to intrusion detection, having as main advantages the components of safety, the comfort with which are executed the commands and the energy rationalization.

This thesis proposes the development of a home automation system for real time integrated monitoring and control of the several technologies, existing in a residential environment.

The developed work consisted in the creation of a home automation network combining a wired medium (twisted pair) with a wireless medium (radio frequency). Using the application layer protocol, MODBUS, it were applied, in the network, microcontrolled modules capable to perform monitoring and actuation functions. It was adopted a centralized architecture, using the SCADA software Movicon, which made it possible to remotely access the system, to provide WEB services and to integrate devices from different manufacturers, due to its compatibility with several communication protocols. Graphically, the developed application, based on the sampled information and on the created databases, is able to present the current state of the devices properties and generate animations according to the user interaction with the system functionalities.



# Conteúdo

1.	Introdução .....	1
1.1	Contexto e motivação .....	1
1.2	Problema .....	1
1.3	Objetivos .....	2
1.4	Organização da dissertação .....	2
2.	Estado da arte .....	3
2.1	X10.....	3
2.1.1	Estrutura das mensagens .....	4
2.1.2	Lista de comandos.....	5
2.2	KNX.....	5
2.2.1	Recursos de configuração .....	6
2.2.2	Estrutura das mensagens .....	7
2.2.3	Definição da interoperabilidade.....	8
2.3	BACnet.....	9
2.3.1	Objetos .....	11
2.3.2	Propriedades .....	12
2.3.3	Serviços.....	13
2.3.4	Estrutura das mensagens .....	14
2.4	C-Bus.....	15
2.4.1	Estrutura das mensagens .....	16
2.4.2	Serviços de aplicação.....	17
2.5	CEBus.....	17
2.5.1	Estrutura das mensagens .....	18
2.5.2	Serviços de aplicação.....	19
2.6	LonWorks .....	20
2.6.1	Estrutura das mensagens .....	22
2.6.2	Serviços de aplicação.....	23
2.7	Universal Powerline Bus.....	24
2.7.1	Estrutura das mensagens .....	25
2.7.2	Serviços de aplicação.....	26
2.8	Comparação dos protocolos .....	27
2.9	Soluções académicas propostas.....	29

2.10	Soluções comerciais .....	31
2.10.1	mControl.....	32
2.10.2	HomeSeer .....	32
2.10.3	Control4.....	32
2.10.4	Mordomus.....	33
3.	Tecnologias de suporte .....	35
3.1	Protocolo RS-485.....	35
3.1.1	Níveis de tensão .....	36
3.1.2	Resistências de terminação.....	36
3.1.3	Resistências de pull-up e pull-down.....	37
3.2	Protocolo MODBUS.....	38
3.2.1	Endereçamento .....	38
3.2.2	Funções .....	39
3.2.3	Verificação de erros.....	39
3.3	Comunicação via radiofrequência.....	40
3.4	Software Movicon .....	41
3.4.1	Ambiente de desenvolvimento .....	42
3.4.2	Movicon na área de domótica .....	43
4.	Arquitetura da solução proposta .....	45
4.1	Definição da arquitetura .....	45
4.1.1	Estrutura da rede .....	46
4.1.2	Microcontrolador .....	47
4.1.3	Transcetor RS-485 .....	48
4.1.4	Transcetor radiofrequência.....	50
4.1.5	Plataforma central.....	53
5.	Implementação da solução proposta.....	55
5.1	Módulo Central .....	55
5.1.1	Funcionamento do microcontrolador central.....	55
5.1.2	Desenvolvimento da plataforma Movicon.....	58
5.1.3	Protótipo da interface com o utilizador.....	59
5.2	Módulos externos .....	67
5.2.1	Módulos RS-485 .....	67
5.2.2	Módulos RF.....	69
5.2.3	Implementação do protocolo MODBUS.....	71

5.2.4	Descodificação do comando IR .....	72
5.2.5	Placas protótipo .....	73
5.2.6	Módulo de medição de energia .....	74
6.	Conclusão .....	75
	Bibliografia .....	77
	Anexo 1 - Orçamento das licenças Movicon .....	81
	Anexo 2 - Esquema elétrico do módulo central .....	83
	Anexo 3 - Esquema elétrico do módulo RS-485 .....	84
	Anexo 4 - Esquema elétrico do módulo RF .....	85
	Anexo 5 - Esquema elétrico da placa protótipo .....	86
	Anexo 6 - Orçamento da placa protótipo.....	87





## Lista de Figuras

Figura 2.1 - Seletores de endereço num módulo de lâmpada X10 [2].	3
Figura 2.2 - Formato da trama X10 [3].	4
Figura 2.3 - Topologia de rede KNX [6].	6
Figura 2.4 - Exemplo de uma rede BACnet [12].	10
Figura 2.5 - Objetos tipo BACnet [14].	11
Figura 2.6 - Serviços BACnet [14].	13
Figura 2.7 - Exemplo de uma rede C-Bus [18].	15
Figura 2.8 - Formato da trama C-Bus série ponto-a-multiponto [18].	16
Figura 2.9 - Exemplo de uma rede CEBus [21].	18
Figura 2.10 - Exemplo de uma rede LonWorks [24].	21
Figura 2.11 - Método de modulação de pulsos UPB [26].	24
Figura 2.12 - Formato da trama UPB [26].	25
Figura 2.13 - Solução proposta pelo autor [28].	29
Figura 2.14 - Solução proposta pelo autor [29].	30
Figura 2.15 - Solução proposta pelo autor [30].	30
Figura 2.16 - Solução proposta pelo autor [31].	31
Figura 3.1 - Comprimento máximo do cabo em função da velocidade de comunicação [37].	35
Figura 3.2 - Níveis lógicos de tensão RS-485 [36].	36
Figura 3.3 - Método de terminação bidirecional RS-485 [38].	37
Figura 3.4 - Resistências de <i>pull-up</i> e <i>pull-down</i> RS-485 [36].	37
Figura 3.5 - Descrição e identificação das funções públicas MODBUS [40].	39
Figura 3.6 - Tipos de modulação de uma onda de radiofrequência [42].	41
Figura 3.7 - Arquitetura de um sistema desenvolvido no Movicon [44].	42
Figura 3.8 - Ambiente de desenvolvimento do Movicon.	42
Figura 3.9 - Lista de <i>drivers</i> de comunicação do Movicon BA.	44
Figura 4.1 - Diagrama de blocos da solução proposta.	45
Figura 4.2 - Pinos do PIC18F26J50 [47].	48
Figura 4.3 - Diagrama e placa de desenvolvimento do transceptor RS-485 utilizado [48].	49
Figura 4.4 - Transceptor de radiofrequência nRF24L01+.	50
Figura 5.1 - Módulo central do sistema desenvolvido.	55
Figura 5.2 - Fluxograma do funcionamento do microcontrolador central.	56
Figura 5.3 - Fluxograma da rotina de interrupção do microcontrolador central.	57
Figura 5.4 - Procedimento de adição do <i>driver</i> MODBUS no Movicon.	58

Figura 5.5 - Procedimento de adição de uma estação MODBUS no Movicon.....	58
Figura 5.6 - Procedimento de definição de uma variável MODBUS no Movicon. ....	59
Figura 5.7 - Ecrã de autenticação do utilizador.....	60
Figura 5.8 - Ecrã de gestão da rede.....	60
Figura 5.9 - Ecrã de gestão dos utilizadores.....	61
Figura 5.10 - Ecrã principal.....	61
Figura 5.11 - Ecrã de ajuda.....	62
Figura 5.12 - Ecrã de submenus da casa. ....	63
Figura 5.13 - Ecrã de controlo dos sistemas de AVAC.....	63
Figura 5.14 - Ecrã de monitorização dos alarmes. ....	64
Figura 5.15 - Ecrã da entrada. ....	65
Figura 5.16 - Ecrã dos temporizadores.....	65
Figura 5.17 - Ecrã de medição de energia. ....	66
Figura 5.18 - Ecrã de videovigilância. ....	66
Figura 5.19 - Ecrã de tarefas WEB. ....	67
Figura 5.20 - Módulo RS-485 do sistema desenvolvido. ....	68
Figura 5.21 - Fluxograma do funcionamento do Módulo RS-485. ....	68
Figura 5.22 - Fluxograma da rotina de interrupção do Módulo RS-485. ....	69
Figura 5.23 - Módulos RF do sistema desenvolvido.....	70
Figura 5.24 - Fluxograma do funcionamento do Módulo RF. ....	70
Figura 5.25 - Fluxograma da rotina de interrupção do Módulo RF.....	71
Figura 5.26 - Comando IR e recetor IR usados. ....	72
Figura 5.27 - Trama IR que codificava o botão 0 capturada com o picoscope. ....	72
Figura 5.28 - Componentes e circuito impresso da placa protótipo.....	73
Figura 5.29 - Dispositivo de medição de energia WM22-DIN [51].....	74

## Lista de Tabelas

Tabela 2.1 - Descrição e identificação dos comandos X10 [3].	5
Tabela 2.2 - Formato da trama KNX [6].	7
Tabela 2.3 - Descrição e caracterização de alguns dos tipos de dados padronizados KNX [8].	8
Tabela 2.4 - Modelo OSI e camadas equivalentes BACnet [10].	9
Tabela 2.5 - Formato da identificação numérica dos objetos BACnet.	12
Tabela 2.6 - Algumas propriedades de um objeto de dispositivo BACnet [15].	12
Tabela 2.7 - Formato da trama MS/TP BACnet [16].	14
Tabela 2.8 - Identificação das aplicações C-Bus [19].	17
Tabela 2.9 - Formato da trama CEBus [22].	18
Tabela 2.10 - Classes e aplicações de propósito geral CEBus [20].	19
Tabela 2.11 - Classes e aplicações de propósito específico CEBus [20].	20
Tabela 2.12 - Especificações dos meios físicos de transmissão LonWorks [25].	21
Tabela 2.13 - Formato da trama LonWorks [23].	22
Tabela 2.14 - Descrição e identificação dos comandos de configuração UPB [26].	26
Tabela 2.15 - Descrição e identificação dos comandos de controlo UPB [26].	27
Tabela 2.16 - Comparação dos protocolos de domótica apresentados.	27
Tabela 2.17 - Meios físicos de comunicação suportados pelos protocolos.	28
Tabela 3.1 - Estrutura da trama MODBUS modelo RTU [39].	38
Tabela 3.2 - Estrutura da trama MODBUS modelo ASCII [39].	38
Tabela 4.1 - Periféricos do PIC18F26J50 [47].	48
Tabela 4.2 - Descrição dos pinos da placa MOD-RS485 [48].	49
Tabela 4.3 - Descrição dos pinos do transceptor nRF24L01+ [50].	50
Tabela 4.4 - Formato de uma trama Enhanced ShockBurst™ [50].	51
Tabela 5.1 - Associação de variáveis MODBUS-Movicon-Microcontrolador.	59



## Lista de siglas e acrónimos

<b>AVAC</b>	<i>Aquecimento Ventilação e Ar Condicionado</i>
<b>ASCII</b>	<i>American Standard Code for Information Interchange</i> (Código Padrão Americano para Troca de Informação)
<b>ASHRAE</b>	<i>American Society of Heating, Refrigerating and Air-Conditioning Engineers</i> (Sociedade Americana de Aquecimento, Refrigeração e Ar Condicionado)
<b>Bps</b>	<i>Bits per second</i> (taxa de transferência de bits num segundo)
<b>CAD</b>	<i>Computer Aided Design</i> (Projeto Assistido por Computador)
<b>EIA</b>	<i>Electronic Industries Alliance</i> (Aliança de Indústrias Eletrónicas)
<b>HMI</b>	<i>Human Machine Interface</i> (Interface Homem Máquina)
<b>IP</b>	<i>Internet Protocol</i> (Protocolo Internet)
<b>IR</b>	<i>Infrared</i> (Infra Vermelhos)
<b>I<sup>2</sup>C</b>	<i>Inter-Integrated Circuit</i> (Inter Circuito Integrado)
<b>IEC</b>	<i>International Electrotechnical Commission</i> (Comissão Eletrotécnica Internacional)
<b>ISO</b>	<i>International Organization for Standardization</i> (Organização Internacional de Normalização)
<b>I/O</b>	<i>Inputs/Outputs</i> (Saídas/Entradas)
<b>LAN</b>	<i>Local Area Network</i> (Rede de Área Local)
<b>MAC</b>	<i>Medium Acces Control</i> (Controlo de Acesso ao Meio)
<b>OPC</b>	<i>OLE (Object Linking and Embedding) for Process Control</i> (Inclusão e Ligação de Objetos para Controlo de Processos)
<b>OSI</b>	<i>Open Systems Interconnection</i> (Interconexão de Sistemas Abertos)
<b>PC</b>	<i>Personal Computer</i> (Computador Pessoal)
<b>RF</b>	Radiofrequência
<b>RTU</b>	<i>Remote Terminal Unit</i> (Unidade Terminal Remota)
<b>SCADA</b>	<i>Supervisory Control and Data Acquisition</i> (Controlo de Supervisão e Aquisição de Dados)
<b>SPI</b>	<i>Serial Peripheral Interface</i> (Interface Periférica Série)
<b>TCP</b>	<i>Transmission Control Protocol</i> (Protocolo de Controlo de Transmissão)
<b>TIA</b>	<i>Telecommunications Industry Association</i> (Associação da Indústria de Telecomunicações)

<b>TTL</b>	<i>Transistor-Transistor Logic</i> (Lógica Transistor-Transistor)
<b>UDP</b>	<i>User Datagram Protocol</i> (Protocolo de Datagrama de Utilizador)
<b>URL</b>	<i>Uniform Resource Locator</i> (Localizador de Recursos Uniforme)
<b>USART</b>	<i>Universal Synchronous and Asynchronous Receiver and Transmitter</i> (Recetor e Transmissor Síncrono e Assíncrono Universal)
<b>UTP</b>	<i>Unshielded Twisted Pair</i> (Par Entrançado sem Blindagem)
<b>VBA</b>	<i>Visual Basic for Applications</i> (linguagem de programação <i>Visual Basic</i> para Aplicações)

# **1. Introdução**

## **1.1 Contexto e motivação**

A domótica é uma tecnologia que visa a automatização de edifícios. Faz uso do recente crescimento tecnológico na área da eletrónica potenciando a gestão de todos os recursos residenciais, através do controlo e monitorização integrados. Esta tecnologia torna mais cómoda a vida das pessoas por satisfazer de forma simples as suas necessidades de conforto e segurança e também provendo-as de ferramentas de racionalização de energia, que reduzem significativamente a sua fatura anual.

O surgimento da domótica deu-se na década de 70 e o intuito foi o de se controlar a iluminação, a climatização e a segurança nos edifícios. Na Europa a implantação desta tecnologia foi apoiada por programas governamentais apontando, sobretudo, ao uso racional de energia e ao aumento da segurança. Naturalmente, algumas das tecnologias emergentes não prosperaram mas resultaram na adoção de algumas premissas que figuram no contexto corrente da domótica.

No panorama atual, apesar da evolução tecnológica, das funcionalidades e das virtudes de um sistema domótico, estes continuam a não equipar a maioria das habitações, notando-se assim alguma resistência ao crescimento do mercado de domótica.

## **1.2 Problema**

As habitações devem responder às necessidades básicas do ser humano, de proteção e bem-estar. Na busca do aumento de qualidade de vida em habitações convencionais, verifica-se um número crescente de sistemas de automação que, por sua vez, não são integrados entre si. Havendo a possibilidade de se interligarem os vários sistemas, elimina-se a necessidade de fidelização a um só fornecedor. Este processo resultará numa expansão tecnológica, e numa redução no preço dos equipamentos, a médio e longo prazo.

Paralelamente, nos ambientes domésticos, a energia geralmente representa uma percentagem significativa da fatura anual, pelo que a gestão dos recursos energéticos é considerada um dos principais desafios. Assim sendo uma participação ativa dos consumidores, na gestão dos seus consumos, torna-se importante para a diminuição dos impactos ambientais resultantes da produção de energia.

### 1.3 Objetivos

Num contexto de arquiteturas comprometidas ambientalmente e atentas à gestão racional de recursos, esta dissertação foi proposta com o principal objetivo de apresentar uma solução que possibilite a monitorização e controlo, em tempo real, de todos os dispositivos e tecnologias presentes no ambiente residencial. Em prol deste objetivo dever-se-á também determinar hábitos de utilização tendo em vista a melhoria do funcionamento da habitação e a redução dos consumos energéticos.

### 1.4 Organização da dissertação

Esta dissertação encontra-se organizada em 6 capítulos e 6 anexos. No presente capítulo (Introdução) é feita uma apresentação ao leitor do tema em análise descrevendo-se o enquadramento do trabalho, os problemas os quais se procuram resolver e os objetivos aos quais o autor se propõe na elaboração deste trabalho.

No capítulo 2 (Estado da arte) são analisados alguns dos protocolos existentes no mercado de domótica nomeadamente o X10, o KNX, o BACnet, o C-Bus, o CEBus, o LonWorks e o *Universal Powerline Bus*. Neste mesmo capítulo, faz-se uma comparação entre estes protocolos e apresentam-se algumas soluções académicas e tecnologias existentes no mercado.

No capítulo 3 (Tecnologias de suporte) são descritas as tecnologias que suportam a arquitetura proposta, com o intuito de transmitir ao leitor conceitos que facilitem a compreensão dos capítulos seguintes.

No capítulo 4 (Arquitetura da solução proposta) é apresentada a arquitetura proposta através da qual se procura solucionar os problemas enunciados. Apresenta-se detalhadamente o *software* e o *hardware* nos quais a arquitetura está assente, justificando-se a opção pelas mesmas.

No capítulo 5 (Implementação da solução proposta) apresenta-se o sistema desenvolvido, analisando individualmente os seus módulos constituintes e a plataforma central, descrevendo também os serviços e a interface dada ao utilizador.

No capítulo 6 (Conclusão) apresentam-se ilações retiradas da elaboração deste trabalho e apresentam-se também sugestões de possíveis melhorias futuras.



## 2. Estado da arte

Os protocolos de domótica definem meios de comunicação e linguagem que permitem o contacto das múltiplas tecnologias da habitação entre si. Para responder às necessidades do mercado foram desenvolvidos diversos protocolos que são a base das tecnologias de domótica. Neste capítulo são apresentados e analisados alguns dos mais preponderantes protocolos e tecnologias existentes no mercado de domótica.

### 2.1 X10

O X10 é o protocolo mais antigo desenvolvido para aplicações domóticas. Foi desenvolvido em meados dos anos 70 com o objetivo de transmitir dados pela rede de energia elétrica, através da injeção de sinais a uma determinada frequência. Esta tecnologia denomina-se *powerline* e possibilita o uso da já existente cablagem elétrica, resultando numa redução de custos e de dificuldade de instalação [1].

Os produtos X10 são maioritariamente consumidos nos Estados Unidos, devido aos seus preços competitivos e elevada taxa de divulgação. Embora este protocolo seja aberto, em Portugal, a divulgação e a adoção desta tecnologia ainda se encontram em crescimento pelo que o seu custo é deveras mais elevado do que nos Estados Unidos. Não obstante, esta continua a ser uma das tecnologias mais acessíveis no mercado.

Existem dois tipos de dispositivos X10: dispositivos emissores e dispositivos recetores. Não existem dispositivos com a capacidade de responder e confirmar a realização correta de uma ordem, o que é imprescindível num sistema que monitorize o estado da instalação. Todos os dispositivos conseguem receber os sinais que circulam na rede, existindo a necessidade de identificar o dispositivo destinatário. Assim sendo, para endereçar cada mensagem, o protocolo X10 implementa um método que usa 16 códigos de casa (A - P) e 16 códigos de aparelho (1 - 16) e permite endereçar, inequivocamente, 256 (16 x 16) aparelhos. A Figura 2.1 apresenta os seletores rotativos de atribuição de endereço num dispositivo X10.



Figura 2.1 - Seletores de endereço num módulo de lâmpada X10 [2].

### 2.1.1 Estrutura das mensagens

O envio da informação, através da rede elétrica, ocorre sobre uma portadora com uma frequência de 120kHz. Os impulsos têm a duração de 1ms e ocorrem em sincronismo com a passagem por zero da onda sinusoidal. Esta particularidade é usada pelos recetores para saberem quando devem escutar a linha. O valor binário 1 corresponde à presença de sinal de 120kHz e o valor binário 0 corresponde à ausência de sinal.

Devido ao ruído elétrico inerente ao meio de distribuição, os bits de dados nunca são enviados isoladamente, diminuindo a probabilidade deste ser confundido com um sinal válido, desta forma, sempre que é enviado um bit, no semiciclo seguinte é enviado o seu bit complementar, esta validação reduz o ritmo de transmissão a um bit por ciclo. A Figura 2.2 ilustra o formato da trama X10.



Figura 2.2 - Formato da trama X10 [3].

Como podemos observar a trama X10 possui quatro campos distintos, são eles:

Código de início - sequência (1 1 1 0) que identifica o início de uma trama. É enviado sem que cada bit seja seguido pelo seu complemento.

Código da casa - conjunto de 4 bits (e respetivos complementos) que identificam uma casa.

Código do dispositivo ou da função - conjunto de 4 bits (e respetivos complementos) que identificam o número de um dispositivo ou o código da função a executar, dependendo da mensagem.

Identificação de dispositivo ou de função - bit (e respetivo complemento) que identifica se o campo anterior se refere ao código de um dispositivo (bit = 0) ou ao código de uma função (bit = 1).

Podemos também observar que a codificação do endereço de casa e do endereço de dispositivo não é feita num sistema de numeração binário de base 2 uma vez que na trama da imagem o 'A' corresponde ao 0b0110<sup>1</sup> e o '2' corresponde ao 0b1110.

<sup>1</sup> A notação 0b é usada na representação de um número binário.

### 2.1.2 Lista de comandos

Cada trama de dados X10 é normalmente enviada em dois grupos onde o primeiro indica o dispositivo e o último a função a ser executada. A tabela seguinte descreve os comandos disponibilizados pelo X10.

Tabela 2.1 - Descrição e identificação dos comandos X10 [3].

Código	Comando	Descrição
0 0 0 0	<i>All units off</i>	Desliga todos os dispositivos com o código de casa.
0 0 0 1	<i>All lights on</i>	Liga todos os dispositivos de iluminação com o código de casa.
0 0 1 0	<i>On</i>	Liga um dispositivo.
0 0 1 1	<i>Off</i>	Desliga um dispositivo.
0 1 0 0	<i>Dim</i>	Reduz a intensidade luminosa de um dispositivo.
0 1 0 1	<i>Bright</i>	Aumenta a intensidade luminosa de um dispositivo.
0 1 1 0	<i>All lights off</i>	Desliga todos os dispositivos de iluminação com o código de casa.
0 1 1 1	<i>Extended code</i>	Código de extensão (seguem-se-lhe os 8 bytes do código).
1 0 0 0	<i>Hail request</i>	Solicita resposta de dispositivo com o código de casa.
1 0 0 1	<i>Hail acknowledge</i>	Resposta ao comando anterior.
1 0 1 x <sup>2</sup>	<i>Pre-set dim</i>	Permite selecionar 2 níveis pré-definidos de intensidade luminosa.
1 1 0 0	<i>Extended data</i>	Dados adicionais (seguem-se-lhe 8 bytes de dados).
1 1 0 1	<i>Status is on</i>	Resposta indicando que o dispositivo está ligado.
1 1 1 0	<i>Status is off</i>	Resposta indicando que o dispositivo está desligado.
1 1 1 1	<i>Status request</i>	Pedido solicitando o estado de um dispositivo.

Da tabela anterior, as seis primeiras funções são funções básicas e usadas com maior frequência. As funções *Dim* e *Bright* não se restringem apenas à regulação da intensidade luminosa, podendo também ser usadas para o controlo da subida e descida de estores ou o controlo de aquecimento. As funções *Hail Request* e *Hail Acknowledge* são usadas para determinar se é possível comunicar com uma casa vizinha. Caso esta situação se verifique, é necessário usar um código de casa diferente. As funções *Extended Code* e *Extended Data* são funções introduzidas no X10 para possibilitar o envio de mais comandos ou dados adicionais.

## 2.2 KNX

O protocolo KNX é uma iniciativa baseada em 3 protocolos precedentes o EIB (*European Installation Bus*), o BatiBUS e o EHS (*European Home System*), com o objetivo de conceber um protocolo padrão que respondesse aos requisitos das instalações de domótica no âmbito europeu. Esta é uma tecnologia aberta que garante o funcionamento de produtos de diferentes fabricantes, desde que certificados pela associação KNX. Atualmente, os dispositivos KNX representam mais de 80% dos dispositivos de controlo e de construção vendidos na Europa [4].

O KNX possibilita a construção de um sistema modular reconfigurável, que permite, em qualquer altura do desenvolvimento do sistema ou até depois de este estar implementado,

---

<sup>2</sup> Bit *don't care*

acrescentar mais dispositivos e funções ao sistema. Isto por ser um sistema descentralizado em que os dispositivos comunicam entre si sem necessidade de hierarquia e supervisão de rede.

O KNX suporta quatro meios de comunicação distintos que facilitam a adaptação da rede às condições locais e às funcionalidades exigidas. As ligações podem ser feitas por cabo entrançado, radiofrequência, *powerline* ou IP/Ethernet. Os dispositivos ligados através de meios físicos como o par entrançado ou *powerline* retiram a energia para funcionar através do meio e os dispositivos sem ligação física necessitam de fontes de alimentação adicionais [5].

Os meios podem ser interligados entre si formando uma rede distribuída que pode hospedar até 65 536 dispositivos, em que cada endereço individual é constituído por 16 bits. A topologia permite 256 dispositivos numa linha, sendo que as linhas podem ser agrupadas em conjuntos, denominados por áreas, com uma linha principal. A rede poderá ser formada por um máximo de 15 áreas, juntamente com uma linha de *backbone* (linha de área), tal como ilustrado na Figura 2.3.

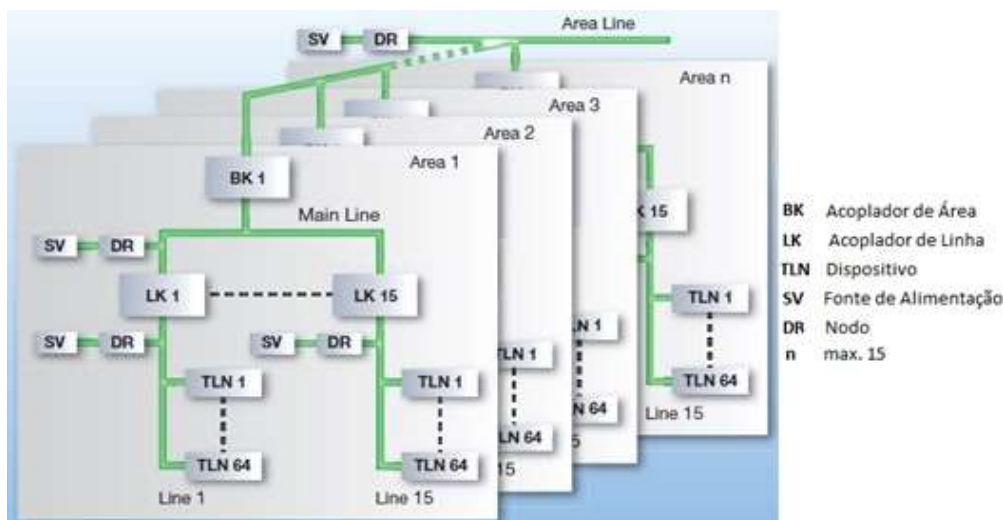


Figura 2.3 - Topologia de rede KNX [6].

### 2.2.1 Recursos de configuração

Para lidar com as diferentes necessidades de configuração, o protocolo KNX inclui um conjunto de recursos que possibilitam a escolha entre vários modos de configuração, cada um adaptado a diferentes mercados, hábitos locais, nível de formação e ambiente de aplicação.

Os modos de configuração dividem-se em *Easy mode* e *System mode* [7]. O *Easy mode* trata-se de um modo de configuração fácil, uma vez que os seus dispositivos já vêm pré-programados para realizar determinada função. No local da instalação os dispositivos são configurados ou por um controlador, ou por botões de pressão ou através de *tags* lógicas (*Controller mode*, *Push-button mode* e *Logical tag extended mode*). Geralmente, os sistemas compatíveis com este modo de configuração têm funcionalidades limitadas e são destinados a instalações de pequenas e médias dimensões. O *System mode* é o modo de configuração mais

utilizado no KNX, pois é o mais versátil e o que permite uma maior adaptação às particularidades da habitação, mantendo a instalação compacta. Destina-se a instalações de maior sofisticação e de grandes dimensões. Neste modo de configuração os dispositivos são instalados e configurados por profissionais certificados, através da ferramenta *Engineering Tool Software* (ETS).

O ETS é o principal recurso de configuração KNX o qual permite, independentemente dos fabricantes, projetar, parametrizar e gerir instalações KNX. O ETS utiliza uma base de dados com a representação gráfica e funcional de todos os dispositivos certificados fornecida pelos fabricantes. Em função destas informações é possível, através de uma interface gráfica, projetar uma rede *offline* passível de ser implementada na prática. Esse projeto será guardado no *software*, juntamente com a informação de configuração dos dispositivos instalados, e poderá ser usado em diagnósticos e resolução de problemas futuros.

Quando ligado à rede, o ETS comporta-se como uma ferramenta *master* de configuração, sob o comando do instalador, permitindo facilmente explorar a instalação e perfilar os diferentes elementos presentes e, a partir dessa informação, ajustar os parâmetros e o comportamento dos dispositivos. Paralelamente, quando combinada com a componente de comunicação IP, a ferramenta ETS permite aceder remotamente à instalação, através de uma ligação IP, ou a partir da LAN local, e efetuar manutenção remota.

### 2.2.2 Estrutura das mensagens

Dependendo do meio, através do qual as mensagens são transmitidas, são adicionados, de acordo com os níveis das camadas do modelo OSI, cabeçalhos ou preâmbulos à trama. De uma forma geral, ignorando estes cabeçalhos e as suas respetivas funcionalidades, o formato base da trama KNX tem a estrutura apresentada na Tabela 2.2.

Tabela 2.2 - Formato da trama KNX [6].

Tabela 2.12 – Formato da trama [6].										
Byte 0	1	2	3	4	5	6	7		8 ... n-1	n ≤ 22
Campo de controlo	Endereço de origem	Endereço de destino			Tipo de endereço	T P C I	A P C I	Dados/ APCI	Dados	Verificação da trama

O campo de controlo determina a prioridade e distingue uma trama padrão de uma trama mais extensa. Uma trama mais extensa pode ter até 255 bytes. O endereço de origem tem 2 bytes e é o do dispositivo emissor da trama. O endereço de destino, também codificado com 2 bytes, pode ser individual (*unicast*) ou de um grupo (*multicast*), a distinção do tipo de endereço é efetuada no campo subsequente. Nesse campo, o primeiro bit indica se o endereço é de um único dispositivo (valor 0) ou de um grupo (valor 1), os 3 bits seguintes representam o *Hop Count*, valor decrementado pelos *routers* de forma a evitar a redundância da mensagem na rede,

e os restantes 4 bits indicam o tamanho do campo de dados. A TPCI (*Transport layer Protocol Control Information*) define as relações de comunicação. A APCI (*Application layer Protocol Control Information*) define os serviços da camada de aplicação que estão disponíveis em função do endereçamento e da relação da comunicação, dependendo do esquema de endereçamento este bloco pode conter um segundo byte. No final da trama é enviado um *Frame check* de um byte que garante a integridade dos dados e possibilita a validação por parte do recetor.

### 2.2.3 Definição da interoperabilidade

A associação KNX define regras para a codificação das informações nas mensagens visando a interoperabilidade de dispositivos de diferentes fabricantes. A codificação de dados tem influência no formato da trama KNX base apresentada anteriormente, tal que os dados codificados em 6 bits ou menos apenas são codificados no campo dados/APCI (os bits não utilizados terão valor 0). Nos restantes tipos de dados é necessário um campo de dados adicional e os 6 bits do campo dados/APCI são definidos com valor 0. A tabela seguinte apresenta alguns destes tipos padronizados de dados e algumas das suas funcionalidades.

Tabela 2.3 - Descrição e caracterização de alguns dos tipos de dados padronizados KNX [8].

Tipos de dados	Tamanho dos dados	Funcionalidades	Descrição
Booleano	1 bit	Interruptores, atuadores, etc.	<i>On/Off, Start/Stop, Verdadeiro/Falso, etc.</i>
3 bits com controlo	4 bits	Controlo de <i>dimming</i> ou persianas	0xxx <sup>b3</sup> diminuir, 1xxx <sup>b</sup> aumentar 001b – 1%      111b - 100%
Caracteres	7 bits	---	Caracteres ASCII
	8 bits		Caracteres ISO 8859-1
Byte sem sinal	8 bits	Porcentagem de um valor, etc.	0 – 0%      255 – 100%
Byte com sinal	8 bits	Contador, etc.	0 – -128      255 – 127
2 bytes com valor <i>float</i>	16 bits	Temperatura, tensão, etc.	-273 °C ... 670,760 °C, -670,760 mV... 670,760 mV
Tempo	24 bits	Dia, hora, minutos e segundos	Dia/Hora - Minutos - Segundos 0bdddhhh - 0bxxmmmmmm - 0bxxssssss
Data	24 bits	Dia, mês, ano	Dia, mês, ano codificados por esta ordem começando pelo byte mais significativo
<i>String</i> de caracteres	14 bytes	---	Conjunto de até 14 caracteres ASCII de 7 bits

A codificação é feita de acordo com a aplicação implementada, e se o tipo de dados estiver padronizado pelo KNX então deve ser adotado pelo fabricante. Na definição de interoperabilidade, além dos tipos de dados para adoção comum, também estão documentados tipos específicos de dados para aplicações de AVAC, para controlo de iluminação, para controlo de persianas e cortinas e para configuração da rede.

<sup>3</sup> Notação binária.

## 2.3 BACnet

O BACnet (*Building Automation and Control Networks*) é um protocolo de comunicação de dados que permite a interoperabilidade entre diferentes sistemas e dispositivos, presentes em aplicações de controlo e de automação residencial. Este protocolo começou a ser desenvolvido em 1987, pela ASHRAE sendo publicado como um padrão ANSI em 1995, e como um padrão global ISO em 2003 [9]. O processo de desenvolvimento deste sistema é aberto sendo o grupo responsável pelo seu desenvolvimento e suporte o ASHRAE *Standing Standard Project Committee* (SSPC).

O BACnet usa uma arquitetura compactada, para aumentar a eficiência da comunicação, através da redução do “excesso de processamento”. Em relação ao modelo de referência, modelo OSI, este apenas implementa quatro das sete camadas, tal como apresentado na tabela.

Tabela 2.4 - Modelo OSI e camadas equivalentes BACnet [10].

Camadas OSI	Camadas equivalentes BACnet						
Aplicação	Camada de Aplicação BACnet						
Rede	Camada de Rede BACnet						
Ligação	ISO 8802-2 type 1	MS/TP	PTP	LonTalk	BVLL	BZLL	
Física	ISO 8802-3	ARCNET	EIA485		UDP/IP	ZigBee	

Podemos observar na tabela que as camadas BACnet de aplicação e de rede são comuns a todos os meios de ligação, o que significa que, independentemente do meio físico de transporte as mensagens de aplicação terão a mesma estrutura. Podemos também observar que o BACnet define 7 meios físicos distintos, através dos quais pretende dar resposta às necessidades das diferentes aplicações [11], nomeadamente:

ISO 8802-3 (Ethernet) - está tipicamente assente em cabos UTP, fibra ótica ou tecnologia *wireless* e consegue velocidades padrão de 10Mbps até 10 Gbps. Equipara-se ao BACnet/IP em termos de custos (altos relativamente aos restantes) e velocidades, mas é limitada a uma única infraestrutura física não sendo encaminhável através de *routers* IP;

BACnet/IP - é implementado sobre a existente infraestrutura Ethernet, adicionando a vantagem de as mensagens poderem ser transmitidas através de redes que utilizem o protocolo internet. Os dispositivos são conectados diretamente a *switches* ou *hubs* de Ethernet e funcionam como um nó IP com o seu próprio endereço IP;

MS/TP (Master-Slave/Token Passing) - usa um protocolo *token passing* implementado por *software*, num microcontrolador, através duma interface série. Tem um custo muito baixo, mas também uma baixa velocidade (9.6Kbps a 115.2Kbps);

ARCNET - usa duas formas de transmissão nomeadamente coaxial a 2.5Mbps, e par entrançado RS-485 a 156Kbps. O meio ARC156K tem um modesto aumento no desempenho em comparação ao MS/TP por uma pequena diferença de custo;

PTP (Point to point) - é usado em redes via linha de telefone, ou para ligação de *modems* sob o protocolo RS-232. Apresenta-se como uma tecnologia com custo moderado sendo as suas características dependentes do *hardware* a conectar;

LonTalk Foreign Frames - permite que a componente de transporte LonTalk seja utilizada para transportar mensagens BACnet, através de par entrançado, *powerline*, fibra ótica ou radiofrequência. O custo desta tecnologia é moderado e as velocidades de transmissão, que são dependentes do transceptor utilizado, variam entre 5kbps, 78kbps e 1250kbps;

ZigBee - é usado em redes sem fios de topologia *mesh* para dispositivos de baixo custo e com limitação na instalação. É a primeira rede sem fios a ser especificada neste protocolo, tem um custo moderado e, na banda de 2.4GHz, permite velocidades até 250kbps.

Numa instalação BACnet, podem ser utilizados diversos meios de transmissão. Para se interligar os diferentes meios são usados *routers* BACnet, que transmitem mensagens entre redes distintas sem alterar ou perturbar o conteúdo da mensagem. Na Figura 2.4 pode-se observar um exemplo de uma rede BACnet com diversos meios de transmissão.

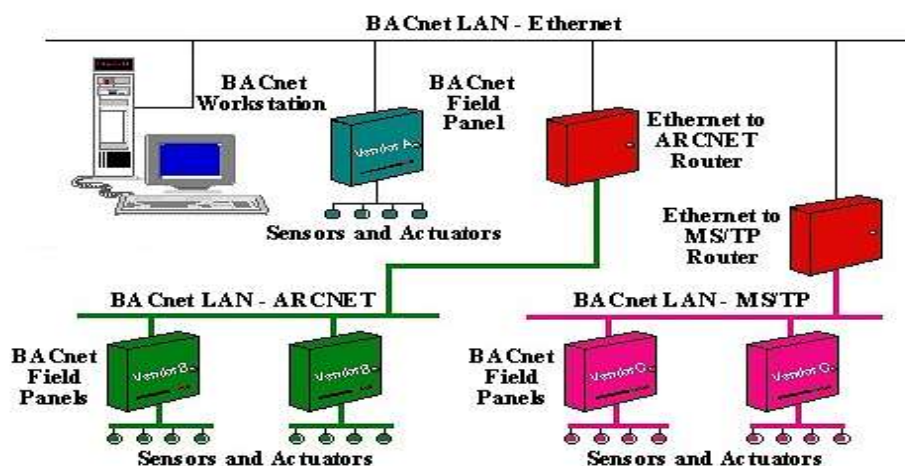


Figura 2.4 - Exemplo de uma rede BACnet [12].

Além dos *routers*, na figura também podem ser observados outros dispositivos BACnet, projetados para compreender e utilizar o protocolo em aplicações específicas. Cada dispositivo BACnet enquadra-se num perfil padrão específico, de forma a garantir as condições mínimas de interoperabilidade. O protocolo, na sua versão mais recente, define oito perfis padrão de dispositivos [13], são eles:

Operator Workstation (B-OWS) - Estação de trabalho de um operador onde se pode monitorizar o sistema mas com capacidades de configuração limitadas;

Advanced Operator Workstation (B-AWS) - Estação de trabalho avançada para controlar e modificar os parâmetros do sistema;

Operator Display (B-OD) - Display simples, que pode exibir dados BACnet para o utilizador final. Normalmente são ecrãs tácteis;



Building Controller (B-BC) - Controlador em que pode ser realizado, de forma extensiva, programação, escalonamentos, gestão de alarmes, registo e partilha de dados com outros dispositivos. Geralmente são controladores de rede ou *routers*;

Advanced Application Controller (B-AAC) - Controlador em que pode ser realizado programação, escalonamentos, gestão de alarmes, registo e partilha de dados, com um avançado grau de complexidade. Geralmente são controladores de campo;

Application Specific Controller (B-ASC) - Controlador de programação e recursos limitados. Geralmente são unidades de volume de ar variável (VAV), de refrigeração ou caldeiras;

Smart Sensor (B-SS) - Sensores capazes de fornecer os dados adquiridos via BACnet;

Smart Actuator (B-SA) - Atuadores com a capacidade de ler e gravar dados via BACnet.

Para se enquadrar num destes perfis, um dispositivo BACnet deve conter um conjunto de informação que o caracteriza. Esta informação subdivide-se em objetos, propriedades e serviços simplificando a representação e gestão dos dispositivos na rede.

### 2.3.1 Objetos

Toda a informação de um dispositivo é representada num ou mais objetos BACnet. Cada objeto representa uma importante componente do dispositivo, ou um conjunto de informações de interesse para outros dispositivos, que traduzem *inputs/outputs* físicos, algoritmos de controlo ou aplicações específicas. O BACnet, na sua versão mais recente, define 54 tipos de objetos distintos para representação e gestão de dispositivos, apresentados na Figura 2.5.

<u>Basic Device Object types</u>	<u>Logging Object types</u>	<u>Meter-related Object types</u>	<u>Simple Value Object types</u>
Device	Event Log	Accumulator	Access Door
Analog Input	Trend Log	Pulse Converter	CharacterString Value
Analog Output	Trend Log Multiple		DateTime Value
Analog Value			Large Analog Value
Binary Input		<u>Presentation-related Object types</u>	BitString Value
Binary Output	<u>Life Safety Object types</u>	Group	OctetString Value
Binary Value	Life Safety Point	Global Group	Time Value
Multi-state Input	Life Safety Zone	Structured View	Integer Value
Multi-state Output	Network Security	<u>Process-related Object types</u>	Positive Integer Value
Multi-state Value		Averaging	Date Value
File		Loop	DateTime Pattern Value
<u>Schedule-related Object types</u>	<u>Physical Access Control Object types</u>	Program	Value
Calendar	Access Point	<u>Notification-related Object types</u>	Time Pattern Value
Schedule	Access Zone	Event Enrollment	Date Pattern Value
<u>Control related Object types</u>	Access User	Notification Class	<u>Lighting Control Object types</u>
Command	Access Rights	Notification Forwarder	Channel
Load Control	Access Credential	Alert Enrollment	Lighting Output
	Credential Data Input		

Figura 2.5 - Objetos tipo BACnet [14].

Associado a cada objeto está um identificador numérico, que combina o número de instância do objeto na rede, com o número que traduz o tipo de objeto. A identificação dos objetos é feita de acordo com a tabela seguinte.

Tabela 2.5 - Formato da identificação numérica dos objetos BACnet.

Byte 0	1	2	3
Tipo de objeto (10 bits)	Instância do objeto (22 bits)		

O BACnet suporta 4 194 304 instâncias de um tipo de objeto, teoricamente. Os objetos de tipo 0 a 127 são objetos BACnet padrão, e de tipo 128 a 1023 são objetos proprietários. Fabricantes independentes podem implementar objetos proprietários, mas para garantirem a interoperabilidade devem seguir o grupo de objetos definido e esta estrutura de identificação. Cada dispositivo BACnet deve ter um Objeto de dispositivo que permite que a sua informação e os seus recursos estejam disponíveis na rede. Ao contrário dos outros objetos, o número de instância do Objeto de dispositivo deve ser único, para cada dispositivo, uma vez que é usado para identificar os dispositivos na rede.

### 2.3.2 Propriedades

O BACnet identifica diversas propriedades padrão para a caracterização e controlo dos objetos. Além destas, os fabricantes podem também definir propriedades proprietárias. Cada tipo de objeto tem um subgrupo de propriedades característico sendo algumas de especificação obrigatória e outras opcionais. A Tabela 2.6 apresenta algumas propriedades, e valores exemplos, de um objeto de dispositivo.

Tabela 2.6 - Algumas propriedades de um objeto de dispositivo BACnet [15].

Propriedade	Valor	Obrigatoriedade
Identificador do objeto	Dispositivo 1076	Necessário
Nome do objeto	Zona 36 DD de controlo	Necessário
Tipo de objeto	Dispositivo	Necessário
Estado do sistema	Operacional	Necessário
Nome do fabricante	Tecnologias Alerton, Inc	Necessário
Identificação do fabricante	Alerton	Necessário
Nome do modelo	VAV-DD Controller	Necessário
Revisão do <i>firmware</i>	1.0	Necessário
Versão do <i>software</i> da aplicação	Dual-Duct DDC	Necessário
Versão do protocolo	1	Necessário
Classe da conformação do protocolo	2	Necessário
Serviços do protocolo suportados	Leitura/Escrita de propriedades, Leitura/Escrita de ficheiros, etc.	Necessário
Tipos de objetos do protocolo suportados	Entrada/Saída Analógica, etc.	Necessário
Lista de objetos	Entrada analógica 1, entrada analógica 2, etc.	Necessário
Comprimento máximo de APDU suportado	50 bytes	Necessário
Suporta segmentação	Não	Necessário

O Identificador do objeto, o Nome do objeto e o Tipo de objeto são 3 propriedades que devem ser definidas em todos os objetos. Cada propriedade contém um identificador de propriedade e o respetivo valor da propriedade. Todas as propriedades na rede devem poder ser lidas e podem, ou não, ser escritas, podem também fornecer comandos. Propriedades proprietárias podem resultar em falhas na interoperabilidade.

### 2.3.3 Serviços

Os serviços são o meio através do qual, um dispositivo adquire informações de um outro dispositivo, comanda um outro dispositivo para executar alguma ação, ou anuncia, a um ou mais dispositivos, o acontecimento de algum evento. Cada solicitação de serviço e reconhecimento do mesmo, traduz-se numa mensagem transferida na rede, a partir do dispositivo emissor para o dispositivo recetor. A topologia é baseada num modelo cliente-servidor em que o cliente requer o serviço e o servidor é responsável por executar o serviço. Na sua versão mais recente o BACnet define 38 tipos de serviços, que estão apresentados na Figura 2.6.

<b><u>Alarm and Event Services</u></b>	<b><u>Object Access Services</u></b>	<b><u>Remote Device Management Services</u></b>
AcknowledgeAlarm	AddListElement	DeviceCommunicationControl
ConfirmedCOVNotification	RemoveListElement	ConfirmedPrivateTransfer
UnconfirmedCOVNotification	CreateObject	UnconfirmedPrivateTransfer
ConfirmedEventNotification	DeleteObject	ReinitializeDevice
UnconfirmedEventNotification	ReadProperty	ConfirmedTextMessage
GetAlarmSummary	ReadPropertyMultiple	UnconfirmedTextMessage
GetEnrollmentSummary		TimeSynchronization
GetEventInformation	<b><u>Virtual Terminal Services</u></b>	UTCTimeSynchronization
LifeSafetyOperation	VT-Open	Who-Has
SubscribeCOV	VT-Close	I-Have
SubscribeCOVProperty	VT-Data	Who-Is
<b><u>File Access Services</u></b>		
AtomicReadFile		
AtomicWriteFile		

Figura 2.6 - Serviços BACnet [14].

As categorias, nas quais se inserem os serviços, têm as seguintes funcionalidades:

Alarmes e eventos - Serviços usados para gerir comunicações relacionadas com notificação de alarmes e eventos;

Acesso a ficheiros - Serviços usados para aceder e manipular ficheiros presentes nos dispositivos;

Acesso a objetos - Serviços usados para aceder e manipular propriedades dos objetos tal como a leitura/escrita de propriedades, criação/eliminação objetos ou a manipulação de listas de dados;

Gestão de dispositivos remotos - Serviços de controlo e monitorização remota de dispositivos;

Terminal virtual - Serviços usados por um utilizador cliente para, estabelecer a conexão com uma aplicação servidora contida num outro dispositivo BACnet.

Os dispositivos não são obrigados a implementar todos os serviços, devem apenas garantir o processamento do serviço *ReadProperty*. A implementação de outros serviços é dependente da função e complexidade do dispositivo, e diz respeito aos fabricantes. Estes deverão definir que objetos e serviços são suportados e perfilar o dispositivo. Toda esta informação deverá ser documentada no Protocolo de Implementação Declaração de Conformidade do dispositivo (*Protocol Implementation Conformance Statement* - PICS).

### 2.3.4 Estrutura das mensagens

As tramas BACnet estão documentadas em dois formatos distintos em função do meio pelas quais são transmitidas, designadamente o MS/TP e o PTP. A nível de informação disponível, apenas as tramas MS/TP têm fontes fidedignas e atuais. Nesta secção serão apresentadas as tramas usadas neste meio cujo formato está apresentado na Tabela 2.7.

Tabela 2.7 - Formato da trama MS/TP BACnet [16].

Tabela 2.7 - Formato da trama IIS/IF-BACnet [16].										
Byte 0	1	2	3	4	5	6	7	... 7 + Tamanho	... + 2	... + 1
Preâmbulo		Tipo de trama	Endereço de destino	Endereço de origem	Tamanho dos dados	CRC do cabeçalho	Dados	CRC dos dados	0xFF (opcional)	
0x55	0xFF									

O preâmbulo, o tipo de trama, os endereços, o tamanho dos dados e o CRC do cabeçalho constituem o cabeçalho da trama. Os dados e o respetivo CRC são designados pelo segmento de dados da trama, e apenas estarão presentes se o campo do tamanho for diferente de 0.

O tipo de trama é usado para fazer distinção entre tramas de diferentes tipos de MAC. Tipos de trama de 8 a 32 e de 34 a 127 estão reservados pela ASHRAE e algumas destas estão disponíveis a organizações, que desenvolvam protocolos abertos, que pretendam implementar diferentes protocolos de rede. As tramas de tipo 32 até 127 são codificadas, e resultam em alterações no formato da trama uma vez que o campo de dados passa a ser codificado e o CRC usado passa a ser o CRC-32K (4 bytes). Tipos de trama de 128 a 255 estão disponíveis para a utilização de tramas proprietárias. O uso destas tramas pode ser implementado para dar suporte a dispositivos que não suportem o BACnet, através de dispositivos híbridos intermediários.

Os endereços de destino e de origem são de um byte cada. O endereço de destino de 255 é o endereço de *broadcast*. Os endereços de 0 a 127 são válidos tanto para nós *master* ou *slave* e os endereços de 128 a 254 são apenas válidos para nós *slave*.

O campo de tamanho especifica o tamanho, em bytes, do campo de dados. O comprimento do campo de dados tem um valor máximo de 501 bytes.

## 2.4 C-Bus

O C-Bus é um sistema de gestão e controlo, para edifícios e residências, baseado em microcontroladores. Este protocolo foi desenvolvido pela empresa Australiana Clipsal, que agora pertence à Schneider Electric, para controlo de iluminação e outros serviços elétricos. Este é um protocolo aberto desde o dia 9 de Dezembro de 2008 e a sua documentação está disponível gratuitamente *online* [17].

Cada dispositivo C-Bus tem um microcontrolador integrado, o que permite que cada unidade seja individualmente programada. Cada unidade transmite a sua atualização de estado em intervalos de tempo específicos, sincronizados através do relógio do sistema, assegurando uma operação rápida e um uso eficiente do meio. Este método descarta o uso de uma plataforma central e permite a transmissão de grandes quantidades de dados num intervalo de tempo pequeno, o que leva a rápidos processamentos e pequenas larguras de banda. Apesar da arquitetura descentralizada e da topologia de rede livre o C-Bus fornece ferramentas que permitem controlo total a partir de um único ponto do sistema, visando a manutenção e resolução de problemas na rede.

Uma rede C-Bus é implementada sobre um cabo de par entrançado, sem blindagem e sem necessidade de terminação na linha. Para cablagens instaladas nos painéis de distribuição é usado um cabo do mesmo tipo mas com revestimento para garantir o isolamento do sinal. O meio não só possibilita a comunicação entre as unidades, como também faculta a alimentação necessária para a operação das unidades. Pode também ser implementada uma rede sem fios bidirecional, através de radiofrequência, paralelamente com o existente meio cablado, através de um *gateway* (ponto que funciona como interligação entre redes).

Uma rede C-Bus pode ser dividida em sub-redes com um máximo de 100 unidades e um comprimento máximo de 1000 metros por sub-rede. Cada sub-rede requer um sistema gerador de relógio para permitir a sincronização de dados. As sub-redes são interligadas através do uso de *bridges* (equipamento de interligação) tal como ilustrado na Figura 2.7.

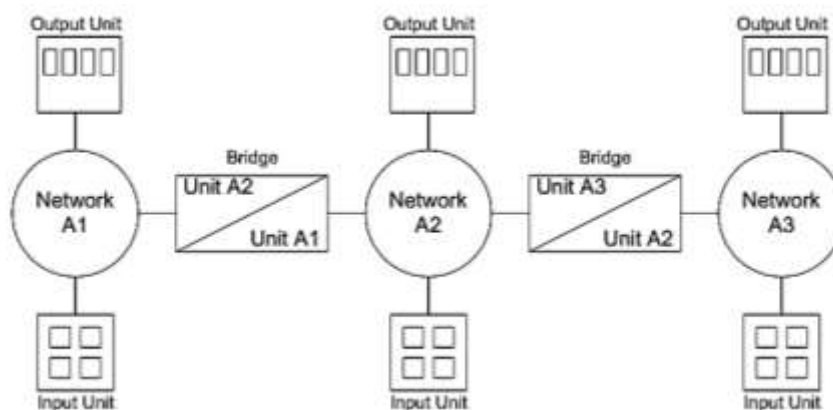


Figura 2.7 - Exemplo de uma rede C-Bus [18].

Esta divisão permite que um sistema C-Bus possa ser gerido por secções, simplificando o projeto, limitando a propagação de potenciais falhas e simplificando a resolução de problemas. Contudo, nas mensagens transmitidas através de *bridges*, é adicionado, através de um protocolo de controlo de informação de rede, o caminho para a rede e o *bridge* de destino.

### 2.4.1 Estrutura das mensagens

As mensagens C-Bus são geralmente direccionadas a uma aplicação invés de um dispositivo específico, desta forma, as mensagens são enviadas para todos os dispositivos e filtradas apenas pelos dispositivos com a aplicação correspondente. A um conjunto de unidades que pertencem à mesma aplicação designa-se por grupo. A estrutura de uma mensagem C-Bus série, com destino a uma aplicação, é a ilustrada na Figura 2.8.

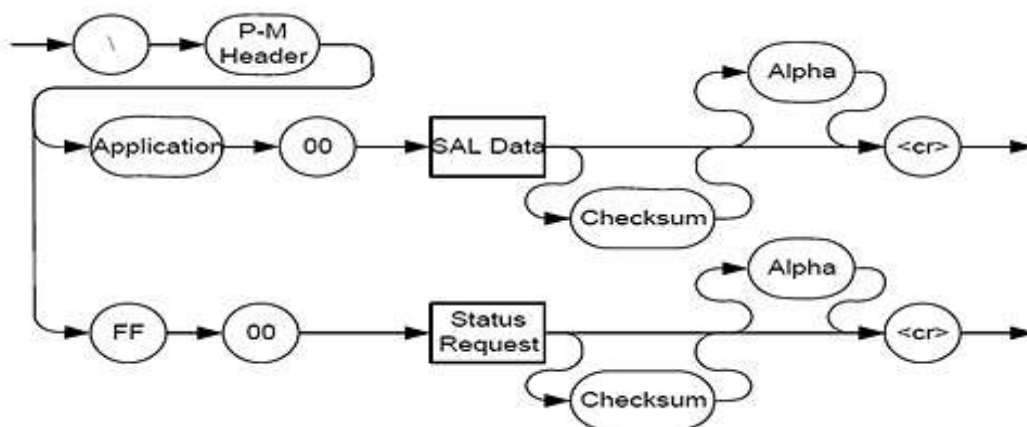


Figura 2.8 - Formato da trama C-Bus série ponto-a-multiponto [18].

Como podemos observar a trama C-Bus série possui os seguintes campos:

Header - é o início de cada mensagem, tem o tamanho de um byte em que os dois bits mais significativos representam a prioridade e os 3 menos significativos representam o tipo de endereço de destino. Os restantes bits são reservados e tem valor 0;

Application - corresponde ao endereço de aplicação e é codificado num byte. Uma unidade pode ter várias aplicações e consequentemente diversos endereços de aplicação;

SAL Data - contém os dados que definem o comando enviado para a aplicação;

Status request - é usado para requisitar o relatório de estado de uma aplicação na rede. Algumas aplicações não suportam esta funcionalidade;

Checksum - é o mecanismo usado para deteção de erros e é codificado num byte. É determinado pelo complemento para dois da soma dos bytes precedentes;

Alpha - é uma letra minúscula entre 'g' e 'z' que, se estiver presente, implica uma resposta de confirmação acerca do estado da transmissão.

Todas as tramas são iniciadas com o carácter '\ ' e terminadas com um *carriage return* ('cr').

## 2.4.2 Serviços de aplicação

A Tabela 2.8 apresenta algumas das aplicações C-Bus. Além destas aplicações existem algumas que não estão aqui documentadas uma vez que não se encontram atualmente no mercado ou estão em conceção.

Tabela 2.8 - Identificação das aplicações C-Bus [19].

Identificação	Aplicação
0x19	Aquisição de temperatura.
0x26	Sistema de controlo do quarto.
0x38 (0x30 - 0x5F)	Iluminação e controlo de carga elétrica.
0x70	Ventilação.
0x71	Sistema de irrigação.
0x72	Controlo de água (piscinas, fontes).
0x73, 0x74	Controlo de atuadores AVAC.
0x88	Aquecimento.
0xAC	AVAC.
0xCA	Controlo de <i>trigger</i> .
0xCB	Ativação do controlo.
0xCE	Reportar erros.
0xD0	Segurança.
0xD1	Contadores de gás, água, eletricidade, etc.
0XD5	Controlo de acessos
0xDF	Relógios e temporizadores
0xE0	Estado e controlo de telefonia.
0xE4	Medição de luz, líquidos, temperatura, etc.
0xF0, 0xFA	Testes <i>Clipsal</i> .
0xFF	Manutenção e controlo da rede.

O número de identificação da aplicação, que podemos observar na tabela, serve para indicar a formatação dos dados na mensagem. Dentro de uma aplicação o formato dos dados deve ser o mesmo, mas pode variar conforme a aplicação.

## 2.5 CEBus

A tecnologia CEBus (*Consumer Electronic Bus*) foi normalizada pelo *Consumer Electronic Group*, da EIA, num esforço de desenvolver uma rede de comunicação para dispositivos de ambiente residencial. Esta iniciativa data de 1984 e resultou na norma EIA-600 publicada em 1995 [20]. Esta norma surgiu devido à incompatibilidade entre os diferentes produtos existentes no mercado, e tinha a finalidade de ser versátil e económica. Permitia o uso de sistemas distribuídos e centralizados e facilitava a instalação em ambientes domésticos de pequenas e grandes dimensões.

O CEBus define os meios físicos de comunicação, os métodos de operação na rede e como é que o dispositivo envia, processa e estrutura a informação. Este protocolo é aberto e implementa as camadas 1, 2, 3 e 7 do modelo OSI. A nível físico suporta *powerline*, par

entrançado, cabo coaxial, infravermelhos, radiofrequência e fibra ótica. Todos os meios implementados transportam as mensagens de controlo a uma taxa média de 8000bps, e, conforme a largura de banda dos meios, podem coexistir dados analógicos ou digitais, de áudio e de vídeo [21].

O CEBus suporta uma topologia de rede flexível, permitindo a fácil adição de dispositivos, e a interligação entre diferentes meios físicos através de *routers*. A Figura 2.9 ilustra um exemplo de uma rede CEBus com três meios distintos interligados por *routers*.

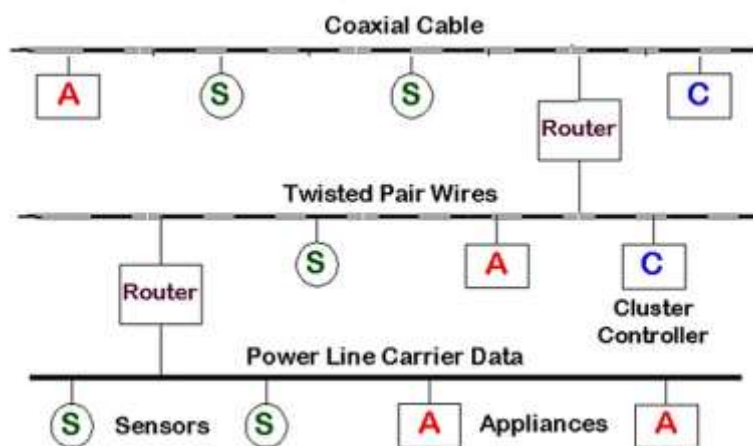


Figura 2.9 - Exemplo de uma rede CEBus [21].

O CEBus não usa um controlador central para gerir a entrega de mensagens. Os seus dispositivos encontram-se todos no mesmo nível de hierarquia e comunicam diretamente entre si. O controlo é distribuído entre os aparelhos CEBus e os *routers*, e todos os dispositivos, de todos os meios, são tratadas como se estivessem no mesmo barramento.

### 2.5.1 Estrutura das mensagens

O formato das mensagens CEBus é independente do meio de comunicação usado. O comprimento das tramas pode variar até um máximo de 44 bytes. A estrutura da trama é a apresentada na Tabela 2.9.

Tabela 2.9 - Formato da trama CEBus [22].

Byte 0	1	2	3	4	5	6	7	8	9	10 - 42	43
Preâmbulo	Controlo	Endereço de destino		Código da casa (destino)		Endereço de fonte		Código de casa (fonte)		Dados	Checksum

A trama CEBus é composta por 8 campos distintos, são eles:

Preâmbulo - campo que contém uma sequência aleatória de 8 bits;

Controlo - campo que contém a prioridade da mensagem, o tipo de trama entre outras informações;

Endereço de destino - contém o endereço do dispositivo destinatário codificado com 2 bytes. Alguns endereços são reservados para *multicast* e *broadcast*;



Código da casa (destino) - codificado em 2 bytes e juntamente com o endereço de destino identifica um nó ou um grupo de nós de destino;

Endereço fonte - identifica o nó ou a zona de origem da mensagem codificado em 2 bytes;

Código da casa (fonte) - codificado em 2 bytes e juntamente com o endereço fonte identifica um nó ou a zona de origem da mensagem;

Dados - contém a informação de aplicação e possui um tamanho máximo de 32 bytes;

Checksum - codificado num byte serve para determinar a integridade da trama.

Como podemos observar os endereços dos dispositivos são codificados em 2 bytes combinados com um código de casa. O código de casa pode ser usado para criar subsistemas numa grande instalação. Em cada subsistema estarão disponíveis 65 536 endereços, embora alguns estejam reservados. Entre os endereços reservados existe uma gama de endereços que é usada para identificar grupos, que permitem a criação de conjuntos lógicos de dispositivos.

O CEBus controla o acesso ao meio através da técnica CSMA/CD CR (*Carrier Sense Multiple Access with Collision Detection and Collision Resolution*), que permite que um dispositivo aceda ao meio de comunicação em qualquer momento desde que este esteja livre. No caso de o canal estar ocupado e existir colisão, os dispositivos têm capacidade de detetar esse facto, sendo assegurado que um deles consegue transmitir a mensagem com sucesso.

## 2.5.2 Serviços de aplicação

Para dar suporte à interação entre as diferentes aplicações, o CEBus define a linguagem CAL - *Common Application Language*. A linguagem é orientada a objetos e enquadra os dispositivos em contexto de propósito geral ou específico. A Tabela 2.10 apresenta algumas das classes e respetivos objetos que modelam aplicações CEBus de propósito geral.

Tabela 2.10 - Classes e aplicações de propósito geral CEBus [20].

Classe	Aplicações
0x05	Interruptores biestáveis
0x06	Sensores biestáveis
0x07	Controlos analógicos
0x08	Sensores analógicos
0x09	Interruptor multi estado
0x0A	Sensor multi estado
0x0B	Interruptores em matriz
0x0C	Medições
0x10	Ecrãs
0x14	Teclados
0x15	Lista de memórias
0x16	Memórias de dados
0x1C	Relógios e temporizadores

Enquanto os objetos de propósito específico são definidos para, suportar as aplicações mais comuns, os objetos de propósito específico são usados para suportar aplicações singulares. A Tabela 2.11 apresenta algumas das classes e respetivos objetos que modelam aplicações CEBus de propósito específico.

Tabela 2.11 - Classes e aplicações de propósito específico CEBus [20].

Classe	Aplicações
0x01	Controlo de nodos
0x02	Controlo de contexto
0x03	Canal transmissor de dados
0x04	Canal recetor de dados
0x11	Transporte do meio
0x13	Discador
0x17	Motor
0x19	Sintonizador
0x1A	Gerador de sons
0x1D	Relógio

## 2.6 LonWorks

O LonWorks foi criado pela empresa Echelon e funciona como base para um sistema aberto e interoperável que visa facilitar a integração de vários dispositivos, de diferentes fabricantes, numa arquitetura e infraestrutura comum. A plataforma é frequentemente apenas referida por LON (*Local Operating Network*) e constitui uma solução generalizada que procura responder aos requisitos dos sistemas de controlo. Esta plataforma é usada nos mercados de iluminação exterior, transporte, serviços públicos, controlo de processos e automação residencial. Os produtos em conformidade com este protocolo são certificados pela organização LonMark de forma a garantir a conformidade com as diretrizes ISO/IEC 14908-1 [23].

Na sua essência as redes LonWorks são construídas a partir de dispositivos de controlo denominados por nós *Neuron Core*. Estes nós possuem uma unidade local de processamento e de gestão dos recursos, o *Neuron Chip*, um transceptor que permite a ligação ao meio físico de transmissão, o *smart transceiver*, e uma interface para ligar subsistemas I/O. O *Neuron Core* inclui até 4 processadores que fornecem capacidade de processamento para comunicação e execução de aplicações. Dois desses processadores são responsáveis pela implementação do protocolo desde a camada de ligação de dados até a camada de apresentação (modelo OSI), um outro é dedicado à implementação da camada de aplicação e o último, presente apenas nas versões mais recentes, gere as interrupções do sistema. Este modelo simplifica o desenvolvimento de dispositivos por parte do fabricante, que apenas se tem de preocupar em desenvolver a aplicação, de forma a correr no *Neuron Core*, e conectar, concordantemente, os subsistemas I/O.

Na Figura 2.10 é apresentada uma estrutura típica de uma rede Lonworks.

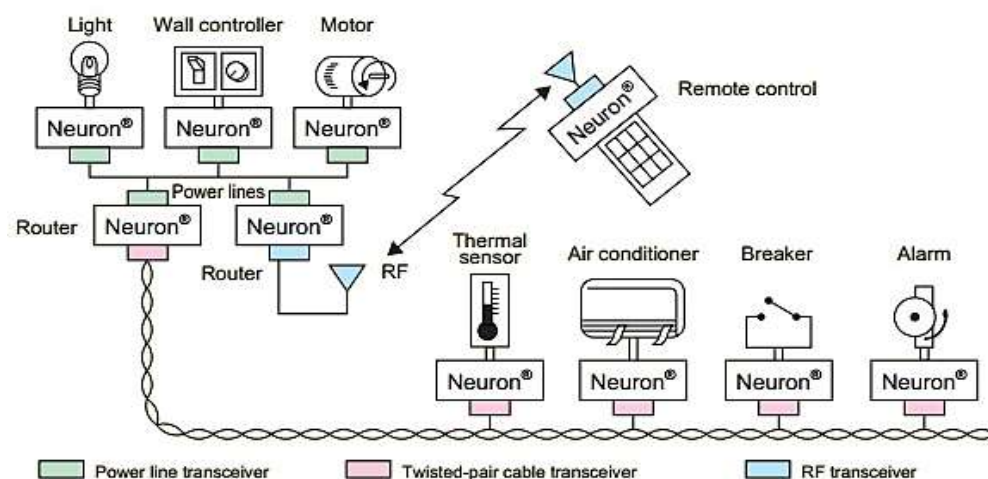


Figura 2.10 - Exemplo de uma rede LonWorks [24].

A nível de endereçamento o Lonworks define uma hierarquia de três níveis usada para endereçar, inequivocamente, um único dispositivo ou um grupo de dispositivos na rede. Na base da hierarquia encontram-se os nós, no meio da hierarquia encontram-se as *subnets* e no topo os domínios. Um domínio representa um conjunto de dispositivos presentes num ou mais meios e pode, por exemplo, ser usado para distinguir duas aplicações distintas num mesmo meio. Num domínio podem existir até 32 385 nós e 255 *subnets*. As *subnets* identificam conjuntos lógicos de nós.

A nível de meios físicos o protocolo dá suporte a comunicações sobre par entrançado, *powerline*, radiofrequência e fibra ótica. As especificações, taxa máxima de transmissão, distância entre dispositivos e topologias suportadas são dependentes do meio de comunicação e são descritas na Tabela 2.12.

Tabela 2.12 - Especificações dos meios físicos de transmissão LonWorks [25].

<i>Transceiver</i>	Meio Físico	Velocidade de transmissão	Topologia da rede	Distância máxima
TP/FT-10	Par entrançado	78 kbps	Livre (comum barramento)	2700 m (barramento) 500 m (livre)
TP/LP-10	Par entrançado	78 kbps	Livre (comum barramento)	2700 m (barramento) 500 m (livre)
TP/XF-1250	Par entrançado	1.25 Mbps	Barramento	130 m
PLT-22	<i>Powerline</i>	Não disponível	Livre (comum barramento)	Quilómetros
LonWorks IP	Canais de internet	Não disponível	Livre	Mundial (internet) Local (LAN)
FO-10	Fibra	1.25 Mbps	Barramento	30 km

### 2.6.1 Estrutura das mensagens

A implementação LonWorks da norma ISO/IEC 14908-1 (*Control Network Protocol - CNP*) para comunicações é o LonTalk. A inclusão do LonTalk no *firmware* do *Neuron Core* é uma das principais características do Lonworks, que, como referido anteriormente, permite ao fabricante abstrair-se dos serviços de comunicação.

Este protocolo permite a autenticação do remetente, a prioridade na transmissão, a deteção de mensagens repetidas, a correção de erros, etc. Suporta também serviços de grande importância na gestão da rede tal como a configuração de *routers* e *bridges*, aceder às estatísticas de comunicação de um nó, iniciar/parar/reiniciar a aplicação de um nó, etc. O algoritmo definido para o acesso ao meio é o CSMA. A estrutura de uma trama de dados LonTalk é apresentada na Tabela 2.13.

Tabela 2.13 - Formato da trama LonWorks [23].

Bit 7	6	5	4	3	2	1	0
Bit-sync (configurable number of 1 bits)							0
Pri	Path	Delta Backlog					
Version		Packet Format		Addr Format		Length	
Addres (3 to 9 bytes)							
Domain (0, 1, 3 or 6 bytes)							
Packet (1 to 230 bytes)							
CCITT CRC-16 (2 bytes)							

Como podemos observar a trama Lonworks contém os seguintes campos:

Bit-sync - é um número variável de bits, com 6 bits no mínimo, que permite aos nós recetores sincronizarem os seus relógios com o relógio do nó emissor;

Pri - campo com 1 bit que indica: 1 - trama prioritária, 0 - trama sem prioridade;

Path - campo que indica o canal a usar, em transctores que estejam conectados a dois canais, tal que: 0 - canal principal, 1 - canal alternativo.

Delta Backlog - especifica um incremento no histórico de registos do canal. Os dispositivos recetores adicionam este valor aos seus registos para terem uma contagem de transmissões da trama;

Version - define a versão do CNP, e tem valor zero para a versão atual;

Format - define o formato do trama tal que: 0 - Trama de transporte, 1 - Trama de sessão, 2 - Trama autenticado, 3 - Trama de apresentação;

Addr Format - formato do endereço que define: 0 - *Broadcast* de uma *subnet*, 1 - Endereço de um grupo (*multicast*), 2 - *Acknowledgement* de uma *subnet*/nó ou grupo, 3 - Neuron ID;

Length - Contém o comprimento do campo *Domain*. É codificado em 2 bits e define valores de 0, 1, 3 ou 6 bytes;

Address - contém o endereço de destino tal como definido no campo *Addr Format*, e pode ter de 3 bytes (formato 0 e 1) até 9 bytes (formato 3);

Domain - contém a identificação de um conjunto de dispositivos num ou mais canais;

Packet - contém os dados a serem transmitidos, é do tipo definido no campo *Fomat* e tem um máximo de 230 bytes;

CCITT CRC-16 - é o mecanismo usado para deteção de erros, é codificado em dois bytes e é determinado pelo polinómio  $x^{16} + x^{12} + x^5 + 1$  calculado sobre a trama completa.

## 2.6.2 Serviços de aplicação

A camada de aplicação define um conjunto de serviços para a configuração e diagnóstico da rede e a implementação de aplicações de controlo padrão. A definição destes serviços advém da necessidade de garantir a interoperabilidade entre dispositivos, de diferentes fabricantes, e de permitir que estes possam ser facilmente instalados e configurados na rede.

Os serviços de configuração e diagnóstico da rede são definidos pela norma ISO/IEC 14908-1, enquanto os restantes são definidos através de perfis funcionais e outras normas publicadas pela LonMark. Os perfis de serviços mais comuns são [23]:

Configuração da rede - conjunto padrão de comandos usados para configurar os atributos de rede de um dispositivo. Esses atributos incluem o endereço na rede.

Diagnóstico da rede - conjunto padrão de comandos usados pelas ferramentas de rede para diagnosticar problemas na rede.

Transferência de ficheiros - conjunto padrão de serviços que permitem a transferência de blocos de dados entre dispositivos e ferramentas de rede. Podem ser transferidos ficheiros com até 2 GB, em tramas de dados de 32 bytes.

Configuração da aplicação - fornece uma interface padrão, baseada nas propriedades de configuração do dispositivo, para configurar o seu comportamento na rede.

Especificação da aplicação - fornece um conjunto de interfaces para documentar as tarefas que um dispositivo executa. Cada tarefa está assente num bloco funcional definido por um conjunto de variáveis de rede e de propriedades de configuração.

Diagnóstico da aplicação - fornece uma interface para testar a aplicação e os blocos funcionais.

Gestão da aplicação - fornece uma interface para ativar/desativar ou corrigir o comportamento de um bloco funcional.

Alarmes - fornece uma interface para um dispositivo reportar alarmes e as respetivas condições de alarme.

Registo de dados - fornece uma interface para os dispositivos guardarem dados em registos. Posteriormente estes poderão ser transferidos para um servidor remoto.

Calendarização - fornece uma interface para a calendarização de eventos baseados, ou no dia da semana, ou numa data específica.

Gestão da hora e da data - fornece uma interface para sincronizar a hora e a data dos dispositivos, numa determinada rede.

## 2.7 Universal Powerline Bus

O *Universal Powerline Bus* (UPB) é um protocolo de comunicação desenvolvido pela *Powerline Control Systems* (PCS) para enviar dados através da linha de alta tensão da rede. Devido ao seu baixo custo e elevada fiabilidade o protocolo de comunicação UPB é ideal para aplicações de comando e controlo no mercado de automação residencial [26].

Este método de comunicação consiste na modulação por posição de pulso (*Pulse Position Modulation* - PPM), sendo adicionados picos de tensão numa de 4 posições predefinidas no semiciclo da onda sinusoidal da rede. Este método de modulação foi inicialmente concebido para redes de 60 Hz, existindo desde Julho de 2007 um acordo entre a PCS e a *Smart Home Ltd* para o desenvolvimento de dispositivos para redes de 220V a 50Hz [27]. A taxa de transmissão tem uma relação de 4b/Hz, numa rede de 60Hz será 240bps e numa rede de 50Hz será 200bps. A Figura 2.11 apresenta as posições predefinidas dos pulsos no semiciclo da sinusoide.

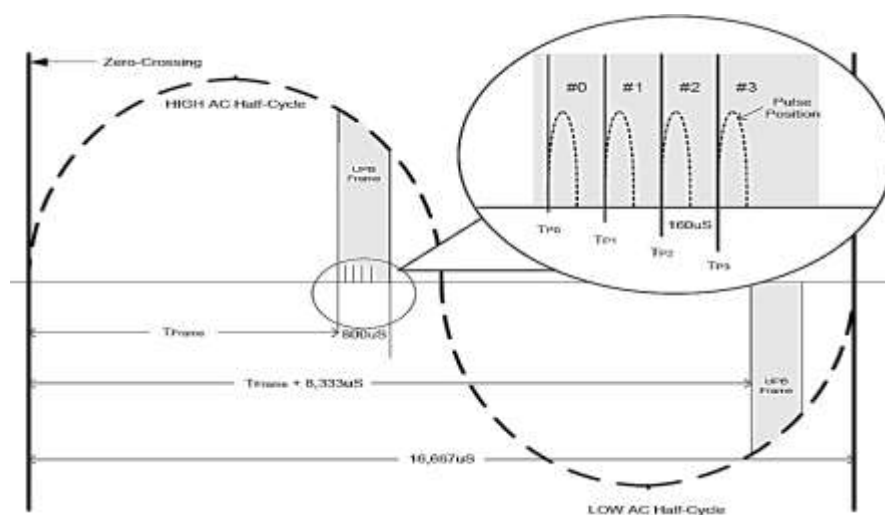


Figura 2.11 - Método de modulação de pulsos UPB [26].

Podemos observar que os tempos de tratamento da informação são todos sincronizados a partir da passagem por zero, de uma transição de um semiciclo negativo para um semiciclo positivo. O intervalo entre pulsos deve ser de 160us sendo que o desvio do pulso da sua posição nominal, para que este possa ser corretamente interpretado, tem de ser inferior a 40us. A posição de cada pulso UPB determina o seu valor como 0, 1, 2 ou 3, ou seja cada pulso consegue codificar 2 bits de dados digitais

### 2.7.1 Estrutura das mensagens

De forma a dar sentido à comunicação, os pulso UPB são organizados em tramas de dados que contêm informação útil para ser transmitida entre dispositivos. Um conjunto de 4 pulsos representa um byte que por sua vez é agrupado num conjunto de 7 a 25 bytes que constituem uma trama de dados tal como estruturado na Figura 2.12.

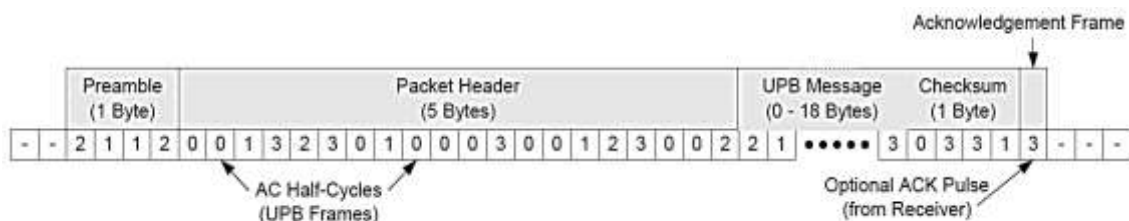


Figura 2.12 - Formato da trama UPB [26].

Como podemos observar o início da trama é definido por um preâmbulo de 1 byte com a sequência de pulsos 2-1-1-2. Este preâmbulo serve para os dispositivos recetores se sincronizarem de modo a receberem a restante mensagem. Os dispositivos recetores podem também mostrar estes pulsos e fazer ajustamentos de acordo com o pico de tensão e a duração.

Posteriormente é enviado um cabeçalho, de 5 bytes, que indica o tamanho da trama, como é que a trama deve ser interpretada, o seu destinatário e o seu remente. O cabeçalho é dividido em 4 campos respetivamente:

Control Word - este campo tem 2 bytes e contém informação que indica o tamanho, a estrutura da trama, de que forma esta deve ser respondida e informação de sequência.

Network ID - este campo tem 1 byte e indica em que rede se encontra o dispositivo destinatário. Embora todos os dispositivos se encontrem no mesmo meio e consequentemente na mesma rede, este campo pode ser usado para definir sub-redes virtuais e formar grupos específicos de dispositivos. Em mensagens de *broadcast* este campo tem valor 0.

Destination ID - este campo é usado para endereçar um dispositivo específico dentro da sub-rede definida. Cada dispositivo tem um endereço de 1 byte, o qual pode ser definido com valores de 1 a 250, sendo os restantes reservados.

Source ID - neste campo está definido o endereço do dispositivo emissor. Este endereço é usado pelo dispositivo recetor nas mensagens que requerem resposta.

Depois do cabeçalho são enviados os dados UPB. Estes são codificados num máximo de 18 bytes, e contém a descrição dos comandos a executar pelo dispositivo recetor. Posteriormente, a fim de verificar a integridade dos dados, é enviado um *Checksum* com um byte, que corresponde ao complemento para dois da soma dos bytes do cabeçalho com os bytes de dados. E por último o pulso de *Acknowledgement*, que apenas é usado pelos dispositivos recetores, para indicar a correta receção da mensagem.

## 2.7.2 Serviços de aplicação

Dentro do campo de dados estão codificados comandos que descrevem as aplicações do protocolo UPB. Nesse campo, o primeiro byte indica a identificação dos comandos e designa-se por *Message data ID* (MDID). Em função do MDID estão documentados comandos que podem ser de configuração, de controlo ou de relatório. Usualmente as mensagens de relatórios são respostas aos comandos de configuração, pelo que de forma a evitar uma redundância de informação apenas serão apresentados os comandos de configuração e de controlo.

Na Tabela 2.14 estão descritos os comandos de configuração, comuns a todos os dispositivos usados pelo protocolo UPB, para configurar os dispositivos e definir o seu comportamento na rede. Estes comandos são identificados com um MDID na gama de 0x00 a 0x1F, pelo que os comandos que se enquadrem nesta gama e não estejam descritos na tabela atualmente não são utilizados, estando reservados para aplicações futuras. O *Link ID*, referido na tabela, é um campo de 1 byte que permite aos dispositivos fazerem a correspondência entre *inputs* físicos de comando e *outputs* físicos de atuação.

Tabela 2.14 - Descrição e identificação dos comandos de configuração UPB [26].

MDID	Comando	Descrição
0x00	Comando nulo.	Não é executada nenhuma ação.
0x01	Ativar escrita	Comanda o dispositivo recetor para desativar a sua proteção de escrita nos registos de <i>setup</i> .
0x02	Proteger escrita	Comanda o dispositivo recetor para ativar a sua proteção de escrita nos registos de <i>setup</i> .
0x03	Iniciar modo <i>setup</i>	Diz ao dispositivo recetor para entrar em modo <i>setup</i> .
0x04	Parar modo <i>setup</i>	Diz ao dispositivo recetor para sair do modo <i>setup</i> .
0x05	Adquirir hora modo <i>setup</i>	Requisita ao dispositivo recetor um relatório de <i>setup</i> com a hora do modo <i>setup</i> .
0x06	Auto endereço	Diz ao dispositivo recetor para, aleatoriamente, selecionar e definir um novo endereço de 1 a 250.
0x07	Obter estado do dispositivo	Requisita ao dispositivo recetor um relatório de estado com o registo de estado.
0x08	Definir o registo de controlo	Define o registo de controlo do dispositivo recetor para um determinado valor.
0x0B	Adicionar <i>link ID</i>	Muda o <i>link ID</i> do dispositivo recetor de “não utilizado” para o <i>link ID</i> especificado.
0x0C	Apagar <i>link ID</i>	Muda o <i>link ID</i> do dispositivo recetor para “não utilizado”.
0x0D	Transmitir esta mensagem	Requisita ao dispositivo recetor para transmitir a mensagem.
0x0E	Reiniciar dispositivo	Requisita ao dispositivo recetor para se reiniciar.
0x0F	Adquirir assinatura do dispositivo	Requisita ao dispositivo recetor o relatório de assinatura.
0x10	Adquirir valor do registo	Requisita ao dispositivo recetor um relatório de registos com um ou mais valores do registo de <i>setup</i> .
0x11	Definir valor do registo	Define o valor de um ou mais registos de <i>setup</i> do dispositivo recetor.



Na Tabela 2.15 estão apresentados os comandos, definidos no protocolo UPB, para enquadrar as aplicações de controlo mais comuns. Estes comandos são identificados por um MDID na gama de 0x20 a 0x3F. Os comandos que se encontram dentro desta gama e não estão descritos na tabela, não são utilizados e estão reservados para aplicações futuras.

Tabela 2.15 - Descrição e identificação dos comandos de controlo UPB [26].

MDID	Comando	Descrição
0x20	Ativar <i>link</i>	Requisita ao dispositivo recetor para ativar os seus componentes de receção <i>linked</i> .
0x21	Desativar <i>link</i>	Requisita ao dispositivo recetor para desativarem os seus componentes de receção <i>linked</i> .
0x22	<i>Goto</i>	Requisita ao dispositivo recetor ou canal para irem para um determinado estado a um determinado ritmo.
0x23	Iniciar desvanecer	Requisita ao dispositivo, de <i>dimming</i> , recetor ou canal para irem para um determinado nível a um determinada ritmo.
0x24	Parar desvanecer	Requisita ao dispositivo, de <i>dimming</i> , recetor ou canal para pararem e ficarem no nível atual.
0x25	Piscar	Requisita ao dispositivo recetor para piscarem a sua saída a um determinado ritmo.
0x26	Indicar	Requisita ao dispositivo, do tipo indicador, recetor ou canal para indicarem um determinado nível.
0x27	Alternar	Requisita ao dispositivo recetor ou canal para alternarem a sua saída a um determinado ritmo, um determinado número de vezes.
0x30	Reportar estado	Requisita ao dispositivo recetor um relatório de estado com a informação do seu estado atual.
0x31	Guardar estado	Requisita ao dispositivo recetor para guardar o seu estado atual na componente de receção endereçada.

## 2.8 Comparação dos protocolos

Neste subcapítulo pretende-se fazer uma comparação dos vários protocolos de domótica, anteriormente descritos. A comparação é feita de forma a poder avaliar os aspetos de maior relevo no mercado e poder identificar os requisitos que carecem uma melhor resposta.

Como tal a comparação centra-se nas propriedades descritas anteriormente nomeadamente os meios físicos, as taxas de transmissão, o endereçamento na rede, as topologias permitidas e os custos. A Tabela 2.16 apresenta algumas destas propriedades para os diferentes protocolos.

Tabela 2.16 - Comparação dos protocolos de domótica apresentados.

Protocolo	Ordem da taxa de transmissão	Endereçamento (dispositivos)	Topologia	Documentação	Custo
X10	bps	256	Barramento	Gratuita	Baixo
KNX	Mbps	65 536	Livre	Paga	Elevado
BACnet	Gbps	4 194 304	Livre	Paga	Elevado
C-Bus	Mbps	256	Barramento	Gratuita	Baixo
CEBus	Mbps	65 536	Livre	Paga	Médio
LonWorks	Gbps	32 385	Livre	Paga	Elevado
UPB	bps	256	Barramento	Gratuita	Baixo

A nível de meios físicos de transmissão, os meios suportados por cada protocolo estão apresentados na Tabela 2.17.

Tabela 2.17 - Meios físicos de comunicação suportados pelos protocolos.

Protocolo	Powerline	Cabo TP	Radiofrequência	Internet Protocol	Outros relevantes
X10	✓	✗	✓	✗	---
KNX	✓	✓	✓	✓	---
BACnet	✓	✓	✓	✓	Fibra ótica
C-Bus	✗	✓	✓	✗	---
CEBus	✓	✓	✓	✗	Fibra ótica
LonWorks	✓	✓	✓	✓	Fibra ótica
UPB	✓	✗	✗	✗	---

Como podemos observar, a partir das tabelas, a tecnologia que mais se evidencia, por ter uma elevada taxa de transmissão, um elevado número de endereços na rede e suporte a diversos meios físicos é o BACnet, em contrapartida trata-se da tecnologia mais dispendiosa tanto a nível de dispositivos como a nível de documentação.

Os protocolos X10 e UPB fazem uso, quase exclusivo, da rede elétrica como meio físico pelo que apresentam taxas de transmissão baixas e uma limitação de endereços na rede. Como vantagens apresentam a fácil instalação e o baixo custo. Estas características tornam estes protocolos os mais indicados em instalações de pequenas dimensões.

Dos restantes protocolos sobressai a inclusão de um microcontrolador nos dispositivos CEBus, C-Bus e LonWorks, que logo à partida dão suporte à comunicação definida pelo protocolo e permitem também que cada unidade seja individualmente programada. Destaca-se também o *software* de configuração ETS, do KNX, que se revela como uma poderosa ferramenta de instalação, configuração e de diagnóstico da rede. Este *software*, embora pese a necessidade de especialização por parte do operador, simplifica bastante a instalação. Também em forma de auxílio ao instalador, salienta-se a organização em objetos, propriedades e serviços definida pelo protocolo BACnet que possibilita a abstração do *hardware* e a independência de fabricantes.

Por último é importante referir que a inclusão do protocolo IP, e o uso da internet como meio de comunicação, revela-se uma grande vantagem uma vez que é para onde se direccionam atualmente as tecnologias. Um fator também importante é a disponibilização da documentação e o facto de um sistema ser aberto, pois leva a que a entidade de desenvolvimento esteja recetiva ao dinamismo inerente à evolução das tecnologias.

Pôde-se concluir que os protocolos distinguem-se pelas condicionantes da instalação às quais pretendem dar melhor resposta, seja o tamanho, as dificuldades ou o custo das instalações. Assim sendo o melhor para o consumidor final será uma infraestrutura que suporte os diferentes protocolos e que através da integração de dispositivos de diferentes fabricantes possa ser um sistema economicamente competitivo.

## 2.9 Soluções académicas propostas

Neste subcapítulo são apresentados estudos no sector da domótica e da automação residencial realizados no âmbito de dissertações de mestrado e de publicações de artigos que retratam o estado da arte neste contexto.

### **“Desenvolvimento de infra-estrutura de comando multifunções EIB-KNX para *Smartphone*” (2012) - Instituto Politécnico de Bragança, Escola Superior de Tecnologia e Gestão [28]**

O problema identificado nesta dissertação foi a necessidade de uma boa conectividade entre elementos heterogéneos numa rede doméstica e o encaminhamento seguro em redes móveis na comunicação com sensores e atuadores. A solução proposta baseava-se numa infraestrutura que permitisse comandar uma instalação de domótica a partir de um *smartphone* tal como ilustrado na Figura 2.13.



Figura 2.13 - Solução proposta pelo autor [28].

A implementação da solução dava suporte a comunicações sem fios, através de redes 802.11 *wi-fi*, comunicações sobre TCP/IP, segundo o protocolo MODBUS/TCP, e protocolos no domínio da domótica nomeadamente o KNX e o KNX-IP. O autor desenvolveu, sobre o sistema operativo *Android*, uma aplicação que, com recurso a uma base de dados, relaciona a estrutura do edifício com a funcionalidade da instalação, e gera uma interface gráfica dinâmica que representa a constituição e as funcionalidades do edifício.

### **“Projeto e construção de um sistema de monitorização de energia elétrica para uma habitação” (2010) - Universidade Nova de Lisboa Faculdade de Ciências e Tecnologia [29]**

Nesta dissertação o autor debruça-se sobre a variante de gestão energética no sector de automação residencial, e a consequente agravante nos impactos ambientais. Como proposta para solucionar este problema foi desenvolvido um sistema de monitorização de energia elétrica, num ambiente residencial, que possibilite a participação ativa dos consumidores na gestão dos seus consumos. A arquitetura da solução está apresentada na Figura 2.14.

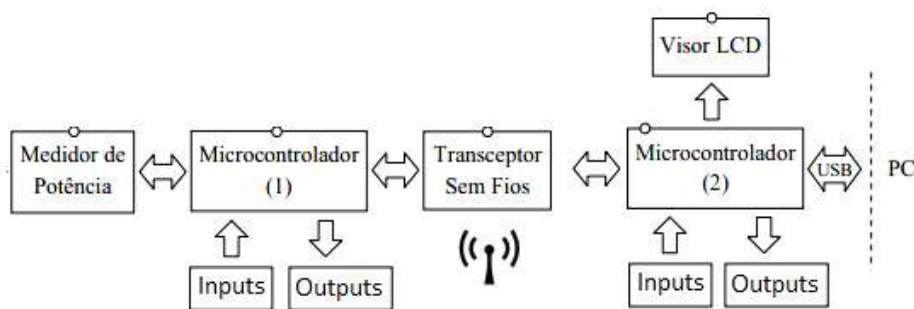


Figura 2.14 - Solução proposta pelo autor [29].

O sistema de monitorização processa a informação em tempo real e através de comunicação sem fios possibilita a amostragem dos dados num LCD (*Liquid Crystal Display*) e o tratamento dos mesmos num PC. Embora a solução implementada apenas retratasse a vertente energética, o uso de 2 microcontroladores e o tratamento de dados no PC permitiria, com uma topologia adequada, a automatização e integração de outras tecnologias residenciais, e uma maior rentabilização do sistema desenvolvido.

### “An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence” (2010) – Universidade de Múrcia [30]

Este é um artigo da autoria de 3 docentes da Universidade de Múrcia que enunciam como desafio a adaptação e a integração das diversas soluções tecnológicas, do mercado de domótica, que respondem ao número emergente de serviços e à variedade das exigências dos clientes. Assim sendo, os autores propõem um novo sistema de domótica que melhora as tecnologias já existentes, considerando as necessidades dos utilizadores e respondendo aos problemas de implantação e integração. A Figura 2.15 ilustra a solução proposta.

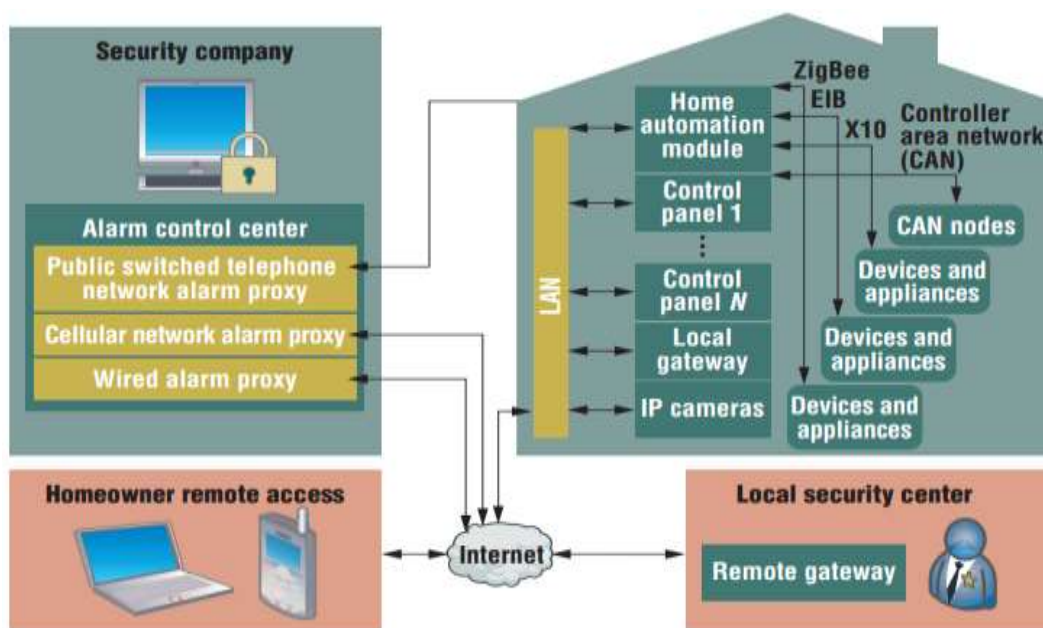


Figura 2.15 - Solução proposta pelo autor [30].

A solução apresentada neste artigo Figura 2.15 integra os protocolos EIB e X10, com tecnologia ZigBee, para monitorizar e controlar os aparelhos domésticos. É também proposto um protocolo de comunicação baseada em IP entre o controlador principal da casa e o resto dos computadores locais e equipamentos remotos.

**“The design and Implementation of a low cost and programmable Home Automation Module” (2006) - Lunghwa University of Science and Technology [31]**

Este artigo baseia-se na premissa de elevados custos e complexa implementação, de módulos domóticos, em redes já existentes. A solução proposta é apresentada na Figura 2.16 e consistiu no desenvolvimento de um módulo de baixo custo com interface remota que possibilitava a gestão simplificada, dos recursos domésticos, ao utilizador.

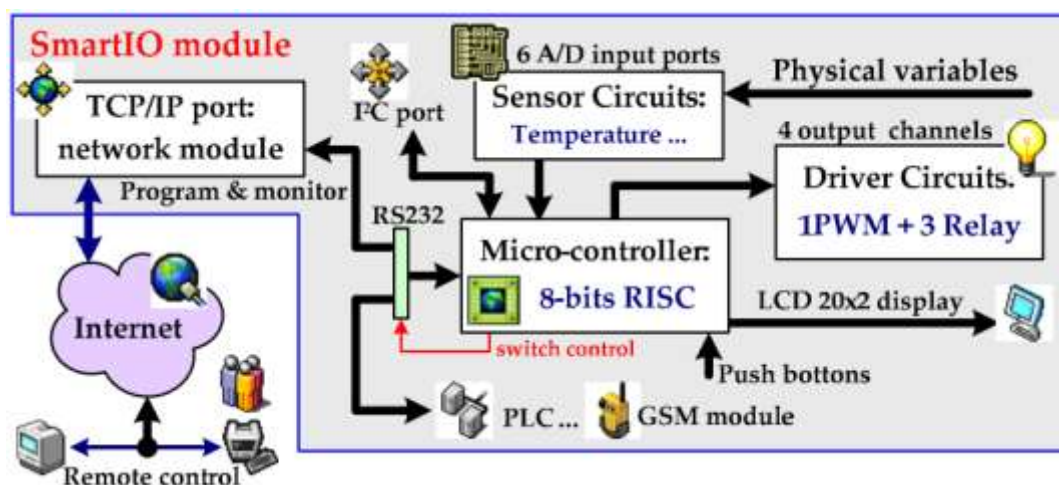


Figura 2.16 - Solução proposta pelo autor [31].

O módulo de domótica foi implementado sobre um microcontrolador de 8 bits que permitia, ao utilizador, programar instruções de controlo tais como temporizadores, funções lógicas, e controlo de atuadores. A comunicação via RS-232 permitiu a conexão a um PLC (*programmable logic controller*) e a um módulo de rede TCP/IP, provendo o sistema de interface remota. Além desta forma de interface o sistema também está equipado com um módulo GSM e um display LCD.

## 2.10 Soluções comerciais

Atualmente existem diversas soluções comerciais no mercado que compõem um vasto leque de sistemas de domótica. Nesta secção pretende-se descrever alguns desses sistemas, escolhidos sobre a base de popularidade entre os consumidores, de forma a enumerar os serviços prestados e descrever as suas mais-valias e debilidades.

### 2.10.1 mControl

O sistema mControl foi desenvolvido pela empresa *Embedded Automation* e encontra-se atualmente na versão 3. O *software* apresenta um custo na ordem do \$170 e funciona em diversas plataformas, nomeadamente computadores pessoais, *smartphones*, *BlackBerrys* e *Windows Phone*. Permite também o acesso remotamente através de qualquer navegador WEB.

O mControl apresenta como pontos fortes a facilidade de uso, o acesso remoto, a programação de tarefas e dispositivos, a compatibilidade com comandos de voz, a compatibilidade com aplicativos de multimédia e a compatibilidade com diferentes protocolos e fabricantes, como o mBee, o KNX, o UPB, o X10, o Z-Wave e o Insteon. As principais limitações, prendem-se com a necessidade de adquirir por terceiros a maioria dos periféricos, ou seja, para usufruir de serviços de gestão de energia ou de segurança e vigilância o utilizador necessita obter os módulos e adicioná-los ao sistema, o que por sua vez pode exigir algum grau de especialização aquando a instalação [32].

### 2.10.2 HomeSeer

O sistema HomeSeer foi projetado para integrar e automatizar os principais sistemas de uma casa tais como iluminação, AVAC, segurança, eletrodomésticos, equipamentos de multimédia entre outros. Permite criar programas de forma a personalizar o funcionamento de qualquer dispositivo, os quais podem ser ativados automaticamente, manualmente ou usando controlo de voz. O *software* tem um preço base de \$249,95.

O HomeSeer fornece acesso remoto, para controlar a habitação a partir de um navegador WEB ou de um *smartphone*, permite monitorizar os padrões de uso de energia, tornando a casa mais eficiente energeticamente, e tem suporte para dispositivos com ou sem fios. Este sistema é compatível com uma grande variedade de *software*, tecnologias de automação e protocolos tais como o Z-Wave, o Insteon, o UPB e o X10. Em relação aos restantes este sistema é o que oferece a melhor seleção de periféricos. O único contra deste sistema prende-se na dificuldade em dominar a interface do utilizador que, embora seja relativamente fácil de usar revela alguns aspetos que carecem melhorias [33].

### 2.10.3 Control4

O sistema de domótica Control4 tem como objetivo fornecer uma maneira atraente e de baixo custo para controlar e automatizar muitas características da habitação. Este sistema permite aos usuários controlar praticamente qualquer dispositivo na casa. O Control4 fornece soluções que incluem equipamentos sem fio, programação, de fácil utilização, e compatibilidade com o protocolo KNX. É possível através de um painel de controlo, de um computador, de um

controlo remoto, ou de um *smartphone* interagir com a habitação numa grande variedade de funcionalidades que incluem o controlo de equipamentos de multimédia, a monitorização da energia, o controlo da iluminação e da segurança. As limitações deste sistema passam pela ausência de controlo por comandos de voz, a necessidade de aderir a um plano pago mensalmente para se obter controlo remoto fora da rede doméstica (4sight) e o facto de os periféricos apenas serem capazes de interagirem com outros da própria marca. Embora numa solução não cablada os dispositivos utilizem tecnologia de comunicação Z-Wave ou ZigBee, o que teoricamente permitiria usar outros dispositivos, o facto de a instalação ser estritamente profissional torna-se difícil de integrar dispositivos de outros fabricantes. A nível de custos, um sistema com as potencialidades base do Control4 tem um custo inicial de \$4000 sendo provido uma garantia de 2 anos [34].

#### **2.10.4 Mordomus**

O Mordomus é um sistema de domótica desenvolvido em Portugal. Este sistema pode ser gerido e controlado local ou remotamente por *software* presente numa consola táctil, possibilitando grande capacidade de monitorização e controlo, de todo o sistema, ao utilizador.

O sistema implementa estratégias de gestão de energia, para um melhor aproveitamento dos períodos energéticos, fornece serviços de vídeo vigilância, soluções *wireless*, com alimentação a partir de energias alternativas, e acesso remoto através de um navegador WEB. O sistema é modular e a empresa que comercializa este sistema desenvolve todos os módulos que podem ser adicionados ao sistema, pelo que o cliente estará sempre dependente da empresa para uma atualização ou mudança no serviço. Embora seja referido no sítio eletrónico, do produto, que este é compatível com outras tecnologias estas não são enumeradas e como referido anteriormente, os módulos são produzidos pela própria empresa, o que porventura indica uma baixa compatibilidade com outros produtos no mercado. A nível de custo um sistema Mordomus está avaliado com um valor médio de cerca 2,5% do valor da casa [35].





### 3. Tecnologias de suporte

A escrita deste capítulo resulta do estudo teórico das tecnologias que, posteriormente, foram adaptadas na implementação do trabalho prático. Serão descritas as tecnologias, que suportam a arquitetura proposta, com o intuito de transmitir ao leitor conceitos que facilitem a compreensão da implementação do sistema.

#### 3.1 Protocolo RS-485

O protocolo RS-485 foi proposto pela TIA/EIA em 1983 e procurava resolver algumas limitações de outros protocolos. A norma TIA/EIA-485 é apenas adotada como especificação da camada física (camada 1 - modelo OSI), podendo ser definido qual o protocolo a ser utilizado para a comunicação dos dados [36].

A norma define um esquema de transmissão de dados que oferece soluções robustas para transmitir dados em longas distâncias e em ambientes ruidosos. A comunicação é efetuada através de um cabo de par trançado de forma a eliminar interferências eletromagnéticas externas que possam causar degradação ou perdas de informação na comunicação. Através de um par de fios, cada dispositivo transmite e recebe dados. Um dispositivo apenas aciona o seu módulo de transmissão no instante em que necessita de transmitir, mantendo-o desligado, o restante tempo, de modo a permitir que outros dispositivos transmitam dados. Apenas um dispositivo pode estar a transmitir, num determinado instante, o que caracteriza esta rede como *half-duplex*. Uma rede RS-485 pode também utilizar 2 pares trançados, permitindo a 2 dispositivos enviar dados simultaneamente operando assim em modo *full-duplex*.

Este protocolo permite interligar até 256 equipamentos, se cada um dos equipamentos apresentar uma resistência interna de  $96K\Omega$ , com uma distância entre equipamentos até 1200 metros. A figura seguinte ilustra a relação entre a velocidade de comunicação e o comprimento máximo do cabo.

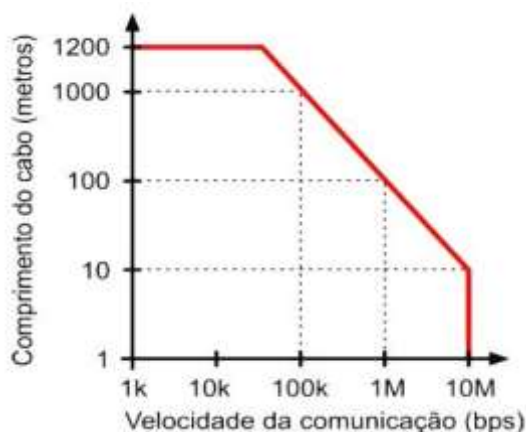


Figura 3.1 - Comprimento máximo do cabo em função da velocidade de comunicação [37].

A velocidade de comunicação depende das características dos equipamentos, da capacidade entre os cabos e das resistências de terminação. Como se pode verificar quanto mais longos os cabos, menor terá de ser a velocidade de comunicação.

### 3.1.1 Níveis de tensão

Como referido anteriormente o protocolo RS-485 caracteriza-se pela utilização de um meio de comunicação diferencial, isto é, os circuitos transmissores e recetores utilizam como informação a diferença entre os níveis de tensão dos condutores do par entrançado. Os códigos binários são identificados pela polaridade da diferença de tensão entre os condutores do par. Quando a tensão no condutor '+' for maior que no condutor '-' (polaridade positiva), é caracterizado um nível lógico 1 e quando, pelo contrário, a tensão no condutor '-' for maior que no condutor '+' (polaridade negativa), é caracterizado um nível lógico 0 (Figura 3.2).

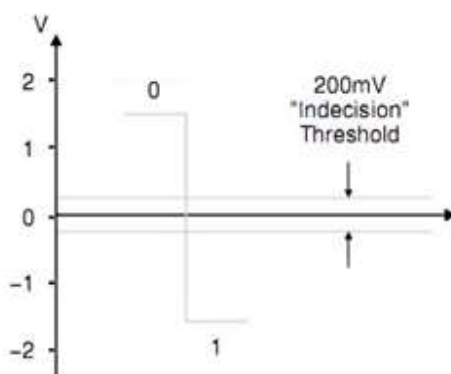


Figura 3.2 - Níveis lógicos de tensão RS-485 [36].

O nível de *threshold*  $\pm 200\text{mV}$  é definido para aumentar a tolerância a interferências. Esta técnica resulta na anulação de ruídos induzidos no meio de transmissão, uma vez que o mesmo ruído é induzido nos 2 condutores, pelo que a diferença de tensão entre eles não se altera e a informação é preservada. A interferência eletromagnética emitida por um barramento de comunicação diferencial é também menor que a emitida por barramentos de comunicação não diferencial.

### 3.1.2 Resistências de terminação

O protocolo descreve a necessidade de terminação nas linhas de comunicação, com uma impedância correspondente ao valor característico da linha de transmissão. A correta terminação atenua reflexões que distorcem os dados transmitidos, aumentando os limites de velocidade e comprimento da rede.

O método de terminação bidirecional (Figura 3.3) oferece uma excelente integridade do sinal. Com esta técnica os dispositivos podem estar localizados em qualquer ponto da rede, a única desvantagem é o aumento do consumo da rede.

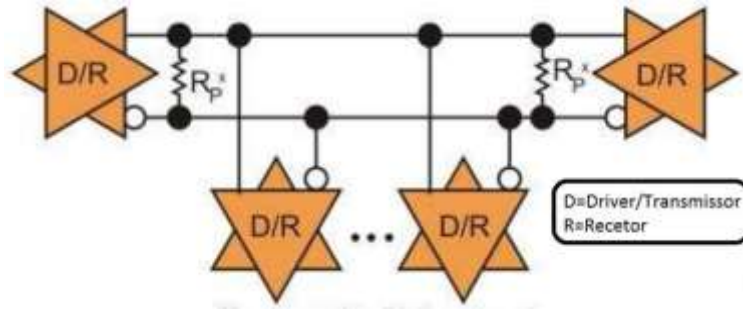


Figura 3.3 - Método de terminação bidirecional RS-485 [38].

As resistências de terminação ( $R_p$ ) devem ser colocadas em paralelo com o par entrançado e dependem das características de impedância do cabo utilizado. A impedância característica de um par entrançado é de aproximadamente  $120\Omega$ , sendo este um valor adequado para as resistências a serem instaladas.

### 3.1.3 Resistências de pull-up e pull-down

O protocolo RS-485 define uma diferença de potencial de 200mV entre os dois condutores do par entrançado, nível de indecisão de *threshold* como referido anteriormente. Para garantir uma diferença de potencial de 200mV entre os dois condutores é necessário utilizar uma fonte de tensão externa  $V_{CC}$  e duas resistências  $R_1$  e  $R_2$  tal como apresentado na Figura 3.4.

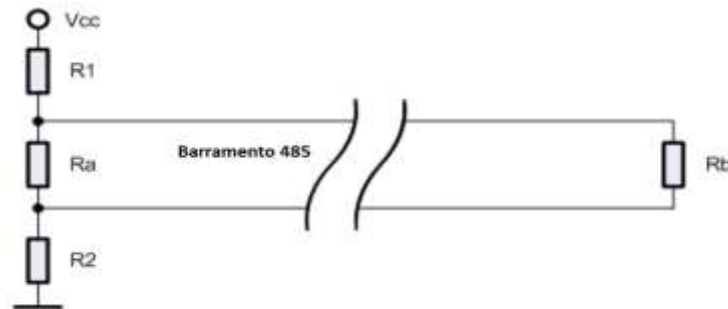


Figura 3.4 - Resistências de pull-up e pull-down RS-485 [36].

As resistências de pull-up e pull-down estarão em série com o paralelo formado pelas duas resistências de terminação ( $R_a // R_b$ ), passando a existir uma corrente entre  $V_{CC}$  e a terra. As resistências de terminação são na ordem dos  $120\Omega$ , para evitar o máximo de reflexões, assim sendo a corrente que atravessa estas resistências, com um *threshold* de 200mV, é:

$$I = \frac{200 \text{ mV}}{R_a // R_b} = \frac{0.2}{60} = 3.3 \text{ mA} \quad (1)$$

Esta corrente percorre as resistências de pull-up e pull-down e cria uma diferença de potencial total de  $V_{CC} - 200\text{mV}$ . Assim sendo o valor da resistência  $R_1 + R_2$  é dado pela expressão:

$$R_1 + R_2 = \frac{V_{CC} - 0.2}{0.0033} \quad (2)$$

Para um valor de  $V_{CC}$  de 3.3V determinou-se o valor para  $R_1 = R_2$  de  $470\Omega$ .

## 3.2 Protocolo MODBUS

O MODBUS foi proposto em 1979 pela empresa Modicon e é um protocolo de mensagens, da camada de aplicação (camada 7 – modelo OSI), que fornece comunicação *Master/Slave* entre dispositivos ligados na mesma rede. Em 2004 este protocolo passou a ser de domínio público tornando-se um dos mais utilizados a nível industrial [39].

Este protocolo define um conjunto de mensagens que todos os equipamentos, que respeitem este protocolo, saberão interpretar e executar, independentemente do tipo de ligações físicas entre os equipamentos. No caso da comunicação série este protocolo define dois modelos de comunicação, são eles o modelo ASCII e o modelo RTU. Nas tabelas seguintes são apresentadas a estrutura das tramas destes modelos.

Tabela 3.1 - Estrutura da trama MODBUS modelo RTU [39].

Endereço (1 byte)	Função (1byte)	Dados (até 252 bytes)	CRC (2 bytes)
-------------------	----------------	-----------------------	---------------

Tabela 3.2 - Estrutura da trama MODBUS modelo ASCII [39].

‘:’	Endereço (2 bytes)	Função (2bytes)	Dados (até 252 bytes)	LRC (2 bytes)	CR+LF
-----	--------------------	-----------------	-----------------------	---------------	-------

Embora a base das duas tramas (Endereço, função, dados e detecção de erros) seja igual a trama do tipo ASCII tem cerca do dobro dos bytes, o que aumenta significativamente o seu tempo de transmissão. A trama modelo ASCII define, também, caracteres de início e de fim de mensagem respetivamente o caracter ‘:’ (caracter ASCII 58) e os caracteres *Carriage Return* (caracter ASCII 13) e *Line Feed* (caracter ASCII 10).

Ao nível do byte, no modelo ASCII são usados 7 bits de dados e no modelo RTU 8 bits, além dos bits de dados é enviado um bit de paridade por cada byte de dados.

O MODBUS usa a notação *big-endian*<sup>4</sup> para endereços e dados. Isto significa que, quando uma quantidade numérica maior do que um único *byte* é transmitida, o mais significativo é enviado primeiro.

### 3.2.1 Endereçamento

O primeiro campo da mensagem é o campo de endereço. Este campo define o acesso ao meio dos *slaves* ligados à rede, isto é, se a mensagem conter o endereço de um determinado *slave* apenas este deverá responder ao pedido efetuado.

O endereço é codificado com 1 byte (modelo RTU) ou 2 caracteres (modelo ASCII). Podem ser codificados até 256 endereços dos quais alguns destes estão reservados. Se a mensagem tiver um endereço de destino 0 destina-se a todos os equipamentos (endereço de *broadcast*),

---

<sup>4</sup> Organização de *bytes* numa memória ou mensagem por ordem decrescente de significância.

os equipamentos que atuam como *slaves* podem ter endereços de 1 a 247 e os endereços de 248 a 255 estão reservados.

### 3.2.2 Funções

O campo Função define o tipo de pedido requerido pelo *master*. Os pedidos dividem-se em leitura/escrita de dados. A função descrita em cada mensagem é codificada em 8 bits (ASCII - 2 caracteres). O protocolo MODBUS define três categorias de funções, nomeadamente funções públicas, funções definidas pelo utilizador e funções reservadas [40].

As funções mais usuais são as públicas, uma vez que as funções reservadas geralmente são propriedade de algumas empresas. As funções definidas pelo utilizador são de acesso livre e podem ser usadas para colmatar a necessidade de funcionalidades específicas que não são traduzidas pelas funções públicas (Figura 3.5).

				Function Codes		
				code	Sub code	(hex)
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02
		Internal Bits Or Physical coils	Read Coils	01		01
			Write Single Coil	05		05
			Write Multiple Coils	15		0F
	16 bits access	Physical Input Registers	Read Input Register	04		04
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03
			Write Single Register	06		06
			Write Multiple Registers	16		10
			Read/Write Multiple Registers	23		17
			Mask Write Register	22		16
			Read FIFO queue	24		18
		Read File record	20		14	
	File record access		Write File record	21		15
	Diagnostics		Read Exception status	07		07
Diagnostic			08	00-18,20	08	
Get Com event counter			11		0B	
Get Com Event Log			12		0C	
Report Slave ID			17		11	
Read device Identification			43	14	2B	
Other		Encapsulated Interface Transport	43	13,14	2B	

Figura 3.5 - Descrição e identificação das funções públicas MODBUS [40].

### 3.2.3 Verificação de erros

Durante a transmissão de mensagens entre equipamentos podem existir erros devido a interferências eletromagnéticas. Para a verificação da integridade dos dados, o MODBUS utiliza o CRC (*Cyclic Redundancy Check*) no modelo RTU e o LRC (*Longitudinal Redundancy Check*) no modelo ASCII.

O algoritmo CRC é similar ao de somas de verificação, embora use a divisão polinomial para determinar o resultado de 16 bits. O CRC permite detetar se um único bit é incorreto pelo que é ideal para prevenir erros aleatórios na transmissão.

A implementação deste algoritmo não é simplista, por ser baseada na análise bit a bit, mas encontram-se disponíveis, publicamente, rotinas que possibilitam a sua aplicação.

No algoritmo LRC o equipamento emissor calcula a soma binária de todos os bytes da mensagem, exceto os bytes correspondentes aos caracteres ':', 'CR' e 'LF', obtendo um resultado limitado a 8 bits. Posteriormente cada bit, desse resultado, é negado sendo incrementada uma unidade ao valor final. Esse valor, de 8 bits, será codificado em dois caracteres e escrito na trama pelo critério *big-endian*.

Estes dois mecanismos resultam no envio de dois bytes adicionais com o valor calculado em função dos bytes da mensagem. Se a mensagem tiver sido transmitida corretamente o valor calculado pelo recetor deverá ser igual ao valor enviado pelo emissor.

### 3.3 Comunicação via radiofrequência

A comunicação via radiofrequência consiste na radiação de sinais eletromagnéticos pelo ar, através dos quais é transmitida energia de um ponto para outro, permitindo uma comunicação sem necessidade de fios. O espectro das ondas de rádio é desde a frequência de 3kHz (muito baixa frequência) até 300GHz (extra alta frequência) [41].

Comumente denominadas apenas por radiofrequência, estas ondas são sinais que se propagam por um condutor físico, normalmente de cobre, e são irradiados no ar através de uma antena. A antena converte o sinal do meio cablado num sinal *wireless* e vice-versa. Os sinais irradiados no ar livre, em forma de ondas eletromagnéticas, propagam-se em diversas direções em função das antenas.

Para transmitir e receber sinais de radiofrequência, entre dois dispositivos, são usados módulos RF que são geralmente pequenos dispositivos eletrónicos. Estes módulos são frequentemente utilizados em produtos para aplicações consumíveis e também em sistemas de automação residencial. As bandas de frequências usadas por estes módulos são designadas por bandas ISM (*Industrial, Scientific and Medical*) que são bandas reservadas, segundo regulamentos, para, como indica a sigla, fins Industriais, Científicos e Médicos. Além das restrições de banda os módulos RF podem também cumprir protocolos de comunicação específicos por exemplo o protocolo Zigbee, Bluetooth ou Wi-Fi.

Na sua essência os módulos RF são constituídos pelo circuito eletrónico de envio e/ou receção, a antena e uma interface através da qual são tratados os dados. Estes módulos dividem-se em três categorias:

Transmissor: Módulo RF capaz de transmitir uma onda de rádio, modulando-a de forma a conter dados. Os transmissores estão sujeitos a regulamentos que ditam a potência máxima de transmissão e os limites de banda.

Recetor: Módulo RF capaz de receber um sinal modulado e interpretar os dados.

Transcetor: Módulo RF que funciona como transmissor ou emissor (*half-duplex*) ou nos dois modos em simultâneo (*full-duplex*).

A informação pode ser incorporada, na onda irradiada, através da modulação de diferentes propriedades, nomeadamente a amplitude, a frequência ou a fase (Figura 3.6). Ao processo de extração de informação de uma onda designa-se desmodulação.

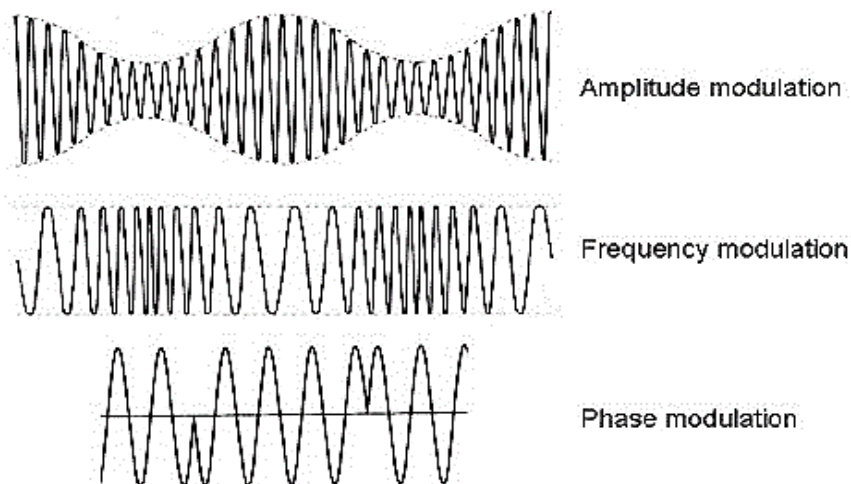


Figura 3.6 - Tipos de modulação de uma onda de radiofrequência [42].

Um fator importante num módulo RF é a potência de transmissão. A potência de transmissão tem influência direta nas condições de funcionamento do módulo. Por exemplo, através do aumento da potência de transmissão, será alcançada uma maior distância de comunicação mas em contrapartida ocorrerá um maior consumo de energia. Além disso, usando uma potência de transmissão mais alta tornará o sistema propenso a interferir com o funcionamento de outros dispositivos. Do mesmo modo, o aumento da sensibilidade do recetor também irá aumentar o alcance de comunicação eficaz, mas irá causar um mau funcionamento devido a uma maior suscetibilidade a sinais provenientes de outros dispositivos RF.

### 3.4 Software Movicon

O Movicon (*Monitoring, vision and control*) é um *software* desenvolvido pela PROGEA direccionado para a criação de interfaces homem-máquina, de monitorização, supervisão e controlo. Este *software* permite, através de diferentes *drivers* de comunicação, interagir com diferentes dispositivos através de diversas interfaces físicas de ligação, garantindo um fluxo de informação em tempo real entre os vários sectores, mesmo num sistema de grandes dimensões. É possível a criação de aplicações para diversas plataformas, desde computadores pessoais, ecrãs tácteis, telemóveis e *tablets* [43].



O Movicon caracteriza-se por ter muita flexibilidade e eficiência em sistemas distribuídos, tal que numa aplicação é possível ter diversas estações em que podem coexistir clientes, servidores e servidores WEB tal como ilustrado na Figura 3.7.

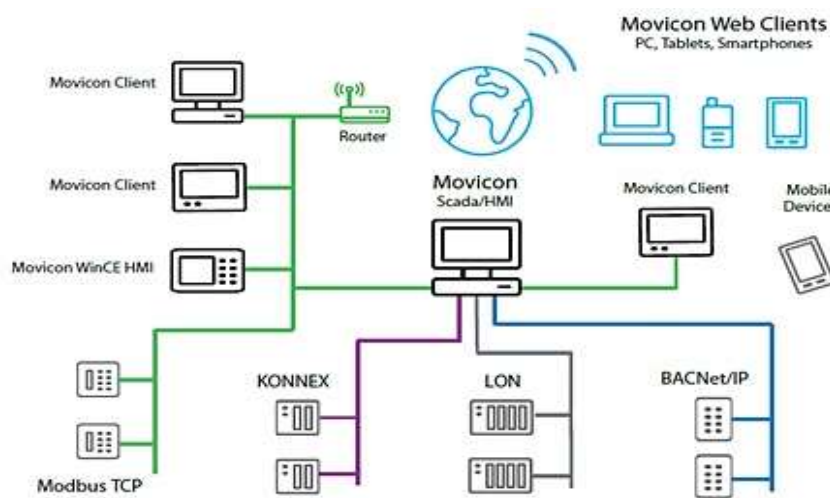


Figura 3.7 - Arquitetura de um sistema desenvolvido no Movicon [44].

### 3.4.1 Ambiente de desenvolvimento

Um projeto Movicon pode monitorizar diversos processos através de janelas ou ecrãs. É possível ter, numa só visualização, sistemas de iluminação, sistemas de segurança, entre outros comumente usados em automação de processos. Num projeto o ambiente de desenvolvimento é orientado a objetos, e dos diversos objetos disponíveis salienta-se a existência de gráficos, tabelas, caixas de texto, mostradores analógicos e ferramentas de gestão de alarmes, de calendarização de tarefas e de registos históricos em base de dados. A interface de trabalho, apresentada na Figura 3.8, é bastante apelativa e intuitiva e permite grande autonomia e fácil autoaprendizagem ao programador.

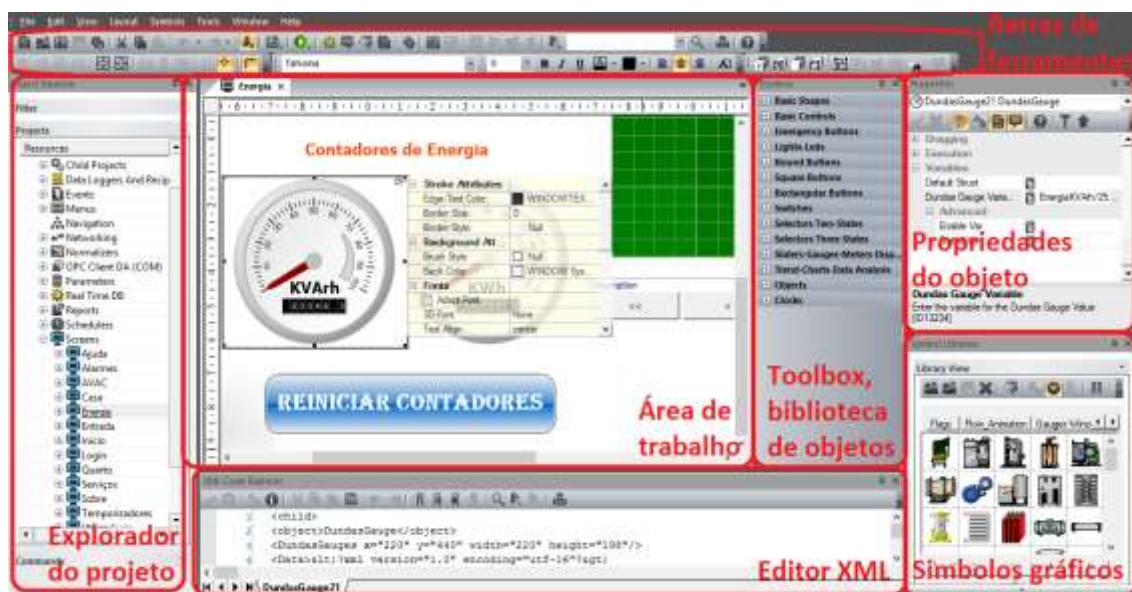


Figura 3.8 - Ambiente de desenvolvimento do Movicon.



Todo o funcionamento de um projeto estará assente em variáveis dinâmicas que podem ser adquiridas pelos diferentes dispositivos conectados. O Movicon permite a definição de imensas propriedades dos objetos com base no estado dessas variáveis. Além disso também permite a criação de *scripts* usando a linguagem VBA, a criação de modelos de OPC através da linguagem *SoftLogic*, e a criação de lógica de controlo através da linguagem visual *Synapsis*. O *software* é baseado em eXtensive Markup Language, XML, um formato de texto simples e flexível, que caracteriza-se pela capacidade de descrever diversos tipos de dados, facilitando a partilha de informações num sistema. A nível de depuração o Movicon permite a simulação do projeto com a inclusão de *breakpoints* e a atribuição de valores, apresentando registos atualizados.

### 3.4.2 Movicon na área de domótica

Dentro do *software* Movicon, a Progea projetou a edição Movicon Building Automation, para gerir instalações e sistemas de edifícios e residências. O *software* Movicon BA oferece as garantias necessárias para a integração de diversos sistemas de gestão de edifícios num mesmo projeto de supervisão e visa responder à crescente demanda da automação residencial. As principais vantagens deste *software* centram-se na gestão de variáveis e bases de dados, oferta de bibliotecas gráficas com animações configuráveis e objetos personalizáveis, ferramentas de gestão e notificação de eventos e alarmes, ferramentas de análise e representação estatística, autenticação e gestão de utilizadores, reconhecimento de comandos de voz e suporte de múltiplas línguas em execução.

Com base no pressuposto de que um sistema de gestão de energia é a base para a deteção de medidas corretivas necessárias para alcançar uma melhoria na eficiência energética, a Progea disponibiliza a ferramenta Pro.Energy que permite identificar os principais indicadores para a redução do consumo de energia e diminuir os custos associados [45]. O Pro.Energy é um módulo funcional, que utiliza a arquitetura do Movicon para comunicação e aquisição de dados e, baseando-se num assistente de configuração, permite conectar uma variada gama de medidores nas diversas fontes de produção e de consumo para medir os dados em tempo real, gravá-los e agregá-los numa base de dados relacional. A partir destes dados esta ferramenta permite facilmente criar relatórios de consumo e de custo em função de intervalos de tempo ou de medidores específicos, gráficos de comportamento, efetuar controlo através da definição de *thresholds*, a criação de grupos lógicos no sistema e o controlo de custos através da personalização do sistema com faixas horárias e tabelas de custos, definidos de acordo com os contratos de energia existentes.

Além disso através de uma única plataforma, sem necessidade de diversos *gateways*, pode ser estabelecida comunicação com diversos protocolos, tais como, protocolos de domótica nomeadamente o BACnet, o KNX e o LonWorks, protocolos usados em unidades de controlo e de segurança como o CEI-ABI, o Elkron, o Guardall PS002, protocolos de gestão de rede tais como o SNMP Manager e outros protocolos de troca de dados, entre eles o MODBUS e o IEC 60870-5 [46]. A lista completa de *drivers* de comunicação fornecidos pelo Movicon BA está apresentada na Figura 3.9.

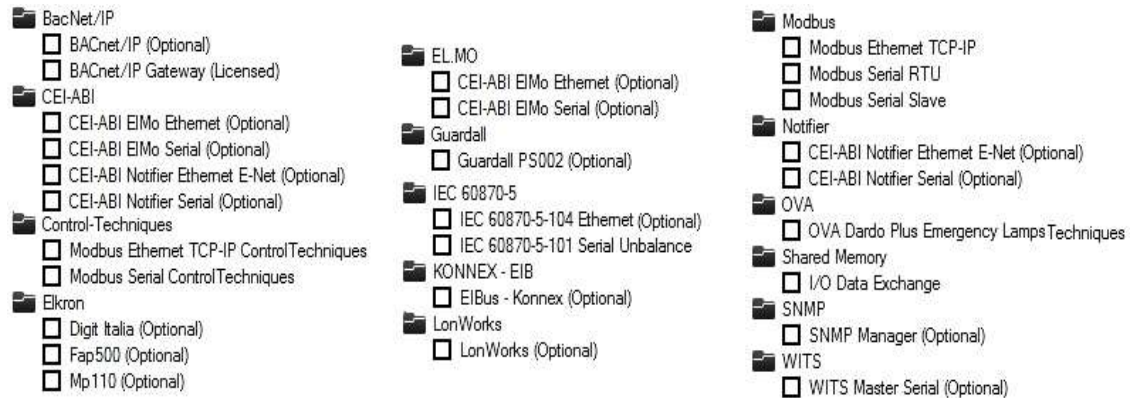


Figura 3.9 - Lista de *drivers* de comunicação do Movicon BA.

## 4. Arquitetura da solução proposta

No presente capítulo descreve-se a arquitetura proposta, a qual visava a resolução dos problemas enunciados inicialmente.

### 4.1 Definição da arquitetura

As tecnologias anteriormente descritas permitiram uma abordagem na conceção de uma arquitetura capaz de cumprir os objetivos propostos. A Figura 4.1 ilustra através de blocos a arquitetura projetada para a solução a implementar.

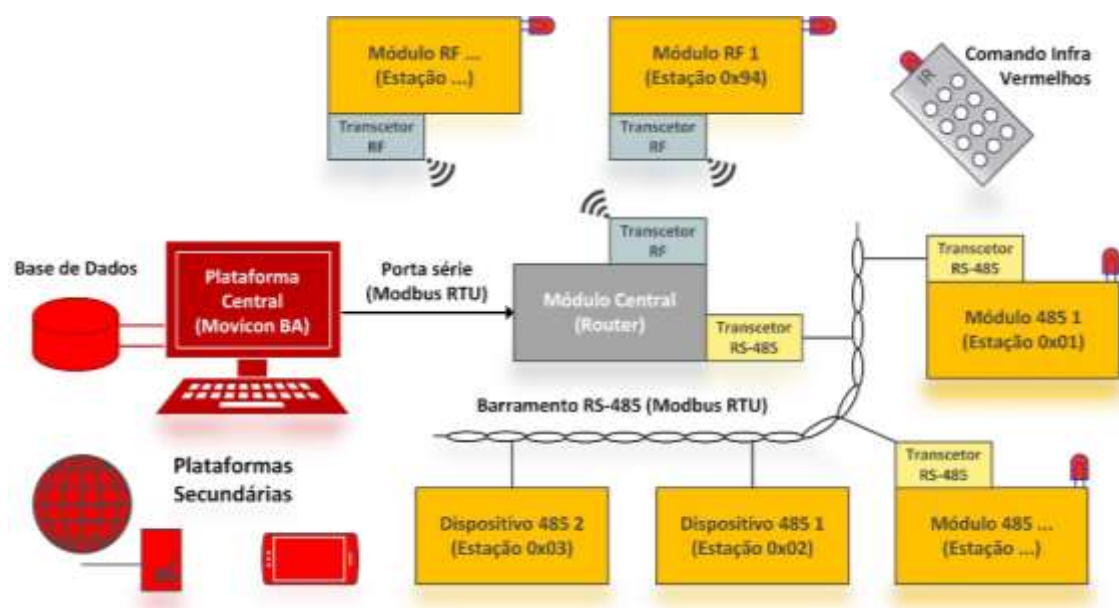


Figura 4.1 - Diagrama de blocos da solução proposta.

Como podemos observar na figura o sistema partia de uma estrutura centralizada, e dividia-se em diversos subsistemas modulares. A plataforma central foi desenvolvida com recurso ao Movicon Building Automation que dava suporte a diversas plataformas secundárias, através de um servidor local, e ao armazenamento de dados devido à compatibilidade com diversas bases de dados. Era através destas plataformas que era dado ao utilizador a interface gráfica de acesso aos serviços de monitorização e de controlo. Esta parte do sistema está representada pelos blocos a vermelho.

Ligado à plataforma central, através de uma interface série, estava um microcontrolador, representado pelo bloco a cinzento, que fazia o reencaminhamento das mensagens no sistema. Este microcontrolador recebia todos os pedidos enviados pela plataforma central e conseguia, através do endereço de destino, determinar em que meio estava localizada a estação de destino. Este microcontrolador, através de dois transceptores, tinha uma interface em cada um dos meios de transmissão e na sua essência funcionava como um *router*.

Os blocos a amarelo representam os subsistemas modulares que, em função do meio onde estão conectados, designavam-se por Módulos RS-485 (barramento RS-485) ou por Módulos RF (rede sobre radiofrequência). Estes módulos podiam ser subsistemas convencionais criados pelo autor que tinham por base um microcontrolador e um transceptor alusivo ao meio em que estavam inseridos, ou então outros dispositivos desde que compatíveis com o barramento RS-485 e cujo protocolo de comunicação seja suportado pelo Movicon<sup>5</sup>. Os módulos convencionais podiam também ser controlados através de um comando de infravermelhos.

#### 4.1.1 Estrutura da rede

Na arquitetura da rede proposta, a opção pelo protocolo RS-485 deveu-se sobretudo aos baixos custos de implementação e ao nível de segurança associado ao meio, uma vez que este apresentava grande imunidade a ruídos e a acessos indevidos por parte de terceiros. Por outro lado, quando comparado com tecnologias sem fios, apresentava maiores requisitos de instalação e limitação na localização dos dispositivos na rede, assim, de forma a dar suporte a um meio de comunicação sem fios, foi proposta a opção de comunicação por radiofrequência.

No meio RS-485, apesar de ser possível a implementação de topologias em estrela ou árvore, a mais recomendada/utilizada é a estrutura em barramento. Como tal, adotou-se a topologia em barramento. Existiam ainda outras considerações a reter na comunicação RS-485, tais como o número máximo de dispositivos, o comprimento máximo da linha, a taxa máxima de transmissão, a impedância de entrada máxima e a utilização de um repetidor de sinal entre cada 32 dispositivos. Dever-se-ia também garantir que os amplificadores operacionais dos nós que efetuavam a transmissão, não aplicassem tensões diferenciais simultaneamente e quando um determinado nó efetuasse a comunicação, todos os outros tivessem os seus amplificadores operacionais em alta impedância, para não influenciarem a linha de transmissão. Todos estes detalhes foram tidos em conta, mas alguns deles devido à dimensão e ao carácter experimental da solução implementada não tiveram preponderância.

No meio RF foi adotada uma topologia em estrela, onde o módulo *master* funcionava como nó central, o qual geria a comunicação. Ambos os meios foram definidos como *half-duplex* e funcionavam segundo um modelo *master/slave*. A definição dos meios como *half-duplex* deveu-se, no caso do barramento RS-485, à necessidade de apenas 1 par de fios entrançados e da compatibilidade de dispositivos *full-duplex* com um meio *half-duplex* (o contrário já não se verificava) e, no caso do meio RF, à maioria dos transceptores existentes no mercado serem *half-*

---

<sup>5</sup> Os protocolos suportados pelo Movicon estão apresentados na Figura 3.9, pg. 44.

*duplex*, uma vez que os transdutores *full-duplex*, por serem mais complexos, são significativamente mais caros.

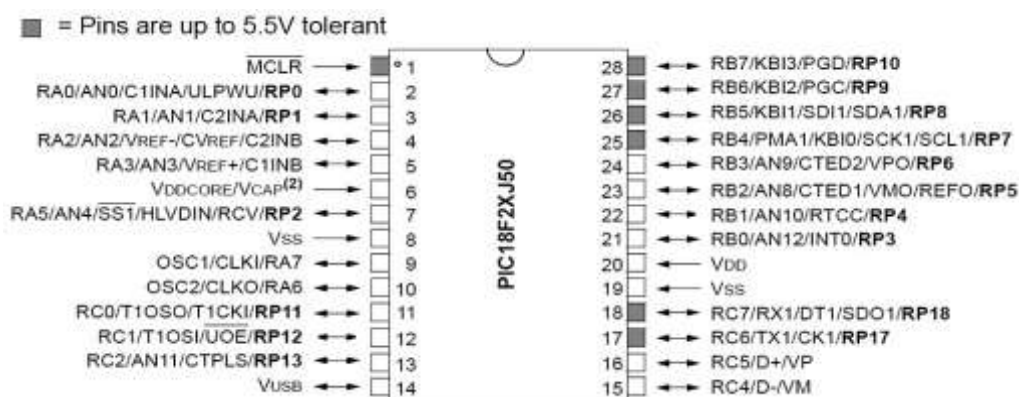
Posteriormente, verificou-se que a principal limitação dos meios era a inexistência de um protocolo que definisse o mecanismo de acesso ao meio e definisse também a camada de aplicação. Como tal e com base no pressuposto que a aplicação de protocolos abertos permitia aos fabricantes abstraírem-se desta parte do desenvolvimento e orientar os seus esforços na potencialização do produto, aumentando também a compatibilidade entre produtos, adotou-se um protocolo com aceitação no ambiente industrial, designadamente o MODBUS.

Como descrito na secção 3.2 (pg. 38) o MODBUS definia a linguagem de aplicação, o acesso ao meio, o endereçamento e a verificação de erros. A nível de endereçamento existiam 256 endereços distintos, mas os únicos endereços que não estavam reservados, e podiam ser atribuídos aos dispositivos na rede, eram a partir do 1 até ao 247 inclusive, ou seja um total de 247 endereços. Então, para se distinguir os dispositivos endereçáveis a partir do barramento RS-485 dos endereçáveis no meio RF, definiu-se os endereços 1 a 147 inclusive como endereços de dispositivos ligados ao meio cablado e os restantes ligados por RF, pelo que a avaliação do destinatário, por parte do microcontrolador do módulo central, era feita com base neste pressuposto. É importante referir que, apesar da limitação de 247 endereços de dispositivos MODBUS na rede, em cada dispositivo MODBUS é teoricamente possível endereçar 65 535 entradas digitais, 65 535 saídas digitais, 65 535 registos de memória de leitura e escrita e outros 65 535 registos de memória apenas de escrita.

#### **4.1.2 Microcontrolador**

Para a conceção dos módulos convencionais e do dispositivo de reencaminhamento de mensagens na rede a escolha recaiu sobre a utilização de microcontroladores. O uso de módulos baseados em microcontroladores, estava já definido por alguns dos protocolos de domótica e além das vantagens claras, de custo e de consumo de energia, alia também vantagens relevantes de flexibilidade de desenvolvimento e de dimensão física. As desvantagens referidas inicialmente na descrição dos protocolos que definem este tipo de arquitetura, eram o uso de um único microcontrolador, apenas comercializado pela entidade responsável pelo desenvolvimento do protocolo, a falta de documentação de suporte e a complexidade da programação desse mesmo microcontrolador. Na arquitetura proposta, estas desvantagens não se verificavam uma vez que poderiam ser usados diferentes microcontroladores com diferentes linguagens de programação, e os protocolos usados, para um dos meios físicos (RS-485) e para a camada de aplicação (MODBUS), são protocolos abertos com documentação gratuita e com

No protótipo desenvolvido pelo autor foi usado o PIC18F26J50 da Microchip, programado sob a linguagem C com recurso ao compilador da Hi-Tech PICC-18 PRO. Este é um microcontrolador de 8 bits de baixa potência da série J dos PIC18F. A Figura 4.2 ilustra os pinos deste microcontrolador.



A nível de periféricos este microcontrolador possuía 1 módulo integrado USB 2.0 que suportava taxas de transmissão até 12Mbps, 2 módulos MSSP (*Master Synchronous Serial Port*) usados para comunicação SPI ou I<sup>2</sup>C <sup>6</sup>, 2 módulos USART (*Universal Synchronous Asynchronous Receiver and Transmitter*) que suportavam os protocolos de comunicação RS-232 e RS-485, 1 ADC (*Analog to Digital Converter*) com 10 canais distintos e uma resolução de 10 bits e ainda 2 módulos ECCP (*Enhanced Capture Compare PWM*) tal como apresentado na Tabela 4.1.

Memória programa	Memória RAM	Pinos	Periféricos de comunicação				ADC	Timers		PWM
64 Kb	3.8 Kb	28	2 USART	2 SPI	1 I <sup>2</sup> C	USB 2.0	10 Bits 10 Canais	2 8 <i>bits</i>	3 16 <i>bits</i>	2 ECCP

Para se criar um meio cablado sob o protocolo RS-485 com o microcontrolador selecionado era necessário adicionar uma transceptor que convertesse os níveis TTL para níveis diferenciais tais como descritos na secção 3.1.1 (pg. 35). As principais características a ter em conta na selecção de um transceptor deste tipo eram a taxa máxima de transmissão, a

bidirecionalidade da comunicação (*half* ou *full-duplex*) entre outras de menor relevância, como tempos de comutação, consumos em modo *shutdown* e impedância de entrada.

Para taxas de transmissão até um máximo de 115 200bps<sup>7</sup> e comunicação *half-duplex* existia uma grande variedade de circuitos integrados que permitiam criar o barramento. Das várias opções existentes a escolha recaiu sob o transceptor RS-485 ADM3483ARZ da *Analog Devices* com recurso à placa de desenvolvimento MOD-RS485 da OLIMEX. Pese o elevado custo da placa de desenvolvimento em relação ao circuito integrado, esta foi apenas usada por uma questão de disponibilidade e de comodidade sendo que o transceptor em si tem um custo de aquisição dos mais baixos no mercado (custo de aquisição de cerca de 0.80€), pelo que uma solução futura passaria apenas pela utilização do circuito integrado.

O transceptor tinha um baixo consumo, funcionava a 3.3V mas interoperável com níveis lógicos de 5V, era *half-duplex* e suportava taxas de transmissão até 250kbps. Os tempos de comutação de estado inativo (alta impedância) para ativo eram na ordem das centenas de nanosegundos. As placas de desenvolvimento MOD-RS485 já continham as resistências de terminação da linha, descritas na secção 3.1.2 (pg. 36), e os conectores que permitiam a ligação ao barramento. A placa e o esquema do transceptor estão apresentados na Figura 4.3.

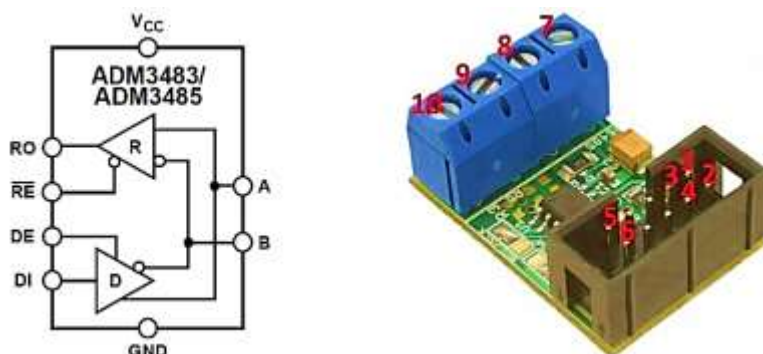


Figura 4.3 - Diagrama e placa de desenvolvimento do transceptor RS-485 utilizado [48].

De acordo com a figura os pinos aos quais se tinha acesso estão descritos na Tabela 4.2.

Tabela 4.2 - Descrição dos pinos da placa MOD-RS485 [48].

Pino	Descrição
1 - VCC	Alimentação (3.3V).
2 - GND	Massa (0V).
3 - DI	<i>Driver Input</i> - Entrada dos dados TTL.
4 - RO	<i>Receiver Output</i> - Saída dos dados TTL.
5 - DE	<i>Driver Enable</i> - entrada digital que permite ativar/desativar o chip.
6 - $\overline{RE}$	<i>Receiver Enable NOT</i> - entrada digital que permite ativar/desativar o modo de receção.
7 - GND <sub>2</sub>	Saída V- (0V).
8 - VCC <sub>2</sub>	Saída V+ (3.3V).
9 - A	Condutor RS-485+.
10 - B	Condutor RS-485-.

<sup>7</sup> A limitação da taxa de transmissão devia-se ao microcontrolador utilizado.

#### 4.1.4 Transcetor radiofrequência

Para se criar um meio sem fios por radiofrequência com o microcontrolador selecionado, era necessário adicionar um transcetor que permitisse enviar e receber mensagens, sobre a forma de sinais eletromagnéticos propagáveis através do ar.

Na seleção deste transcetor devia-se ter em conta aspetos como, a encriptação de dados, a velocidade de transmissão e o alcance da transmissão. As soluções analisadas passavam por circuitos integrados presentes em placas de desenvolvimento as quais apresentavam protocolos proprietários e a possibilidade de serem programadas através de uma interface série. As soluções mais comuns e economicamente mais acessíveis eram baseadas nos circuitos integrados MRF24J40 da Microchip e nRF24L01+ da Nordic Semiconductor. Embora as soluções baseadas no MRF24J40 fossem compatíveis com os protocolos ZigBee™ e MiWi™ [49], enquanto o nRF24L01+ apenas funcionava segundo o protocolo da Nordic o Enhanced ShockBurst™, as primeiras apresentavam custos superiores e velocidades de transmissão inferiores. Como tal o transcetor adotado foi o nRF24L01+ o qual apresentava taxas de transmissão de dados de 250kbps, 1Mbps ou 2Mbps, distâncias máximas de transmissão na ordem das dezenas de metros e um custo significativamente baixo (o preço de aquisição era de cerca de 1€). Na Figura 4.4 é apresentado o transcetor nRF24L01+.

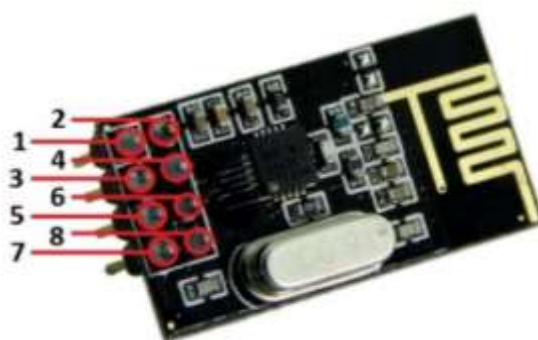


Figura 4.4 - Transcetor de radiofrequência nRF24L01+.

Na Tabela 4.3 são apresentados os pinos do transcetor nRF24L01+ conforme representados na figura.

Tabela 4.3 - Descrição dos pinos do transcetor nRF24L01+ [50].

Pino	Descrição
1 - GND	Massa (0V).
2 - VCC	Alimentação (1.9 a 3.6V).
3 - CE	<i>Chip Enable</i> - entrada digital que permite ativar o chip para receção/transmissão.
4 - CSN	<i>Chip Select NOT</i> - entrada digital que permite selecionar o chip na comunicação SPI.
5 - SCK	<i>Serial Clock</i> - Entrada do sinal de relógio.
6 - MOSI	<i>Master Output Slave Input</i> - Entrada de dados SPI.
7 - MISO	<i>Master Input Slave Output</i> - Saída de dados SPI.
8 - IRQ	<i>Interrupt Request</i> - Saída digital que sinaliza as interrupções (mudança de 1 para 0).



Este transceptor funcionava na banda de 2.4 GHz e caracterizava-se por um baixo consumo, com potências de transmissão de -18, -12, -6 ou 0dbm (ou seja de 16 uW a 1 mW). Como referido o protocolo a nível da camada de ligação de dados denominava-se por Enhanced ShockBurst™ e permitia a assemblagem, o *acknowledgement* automático e a retransmissão, em caso de falha, das tramas. Este protocolo aumentava significativamente a eficiência e a segurança da comunicação, mas limitava a compatibilidade do meio a dispositivos baseados no mesmo transceptor. O formato da trama Enhanced ShockBurst™ é apresentado na Tabela 4.4.

Tabela 4.4 - Formato de uma trama Enhanced ShockBurst™ [50].

<b>Preamble</b> <b>(1 byte)</b>	<b>Address</b> <b>(3-5 bytes)</b>	<b>Packet Control Field</b> <b>(9 bits)</b>	<b>Payload</b> <b>(0-32 bytes)</b>	<b>CRC</b> <b>(1-2 bytes)</b>
------------------------------------	--------------------------------------	--	---------------------------------------	----------------------------------

A trama apresentada era constituída pelo *Preamble* (Preâmbulo) que era usado pelo dispositivo recetor, para se sincronizar com o fluxo de entrada de dados, o *Address* (Endereço) era usado para identificar o dispositivo recetor e podia ser configurado com um tamanho de 3 a 5 bytes, o *Packet Control Field* (Campo de Controlo da Trama) que continha informação sobre a quantidade de dados, a identificação se os dados eram novos ou retransmitidos e a informação sobre a necessidade de envio de um *acknowledgement*, o *Payload* (Dados) que era a mensagem transmitida e o CRC que era o mecanismo de verificação da integridade dos dados e podia ser configurado com 1 ou 2 bytes.

A nível de funcionamento o nRF24L01+ tinha 3 modos de operação que eram:

Power down - estado em que o chip estava inativo pelo que não conseguia nem transmitir nem receber dados. Este estado era usado no modo de repouso a fim de baixar o consumo do chip. Neste modo era na mesma possível efetuar configurações uma vez que a interface SPI continuava ativa;

RX mode - estado em que o chip se encontrava ativo e configurado para receber dados. Neste modo, o chip desmodulava os sinais que se encontravam no canal configurado, e se obtivesse uma trama de dados com o seu endereço e com um CRC válido, guardava os dados numa memória FIFO (*first in first out*) de até 32 bytes. Os dados permaneciam nesta memória até que fossem lidos. Se a memória estivesse lotada os novos dados recebidos seriam descartados;

TX mode - estado em que o chip se encontrava ativo e configurado para enviar dados. Neste modo para se enviar dados devia-se primeiro escrevê-los na memória FIFO de envio e posteriormente, assim que o pino CE tivesse um valor lógico 1 por mais de 10uS, os dados seriam enviados.

Para representar a receção/envio de dados sucedido e a máxima retransmissão automática de dados, em caso de falha, o nRF24L01+ definia interrupções no pino IRQ, que funcionavam da seguinte forma:

1 - Era iniciada a transmissão de uma trama de dados do emissor para o recetor. Após a conclusão do envio o dispositivo emissor ia para *RX\_Mode* e esperava por uma trama de *acknowledgement*.

2 - Se o dispositivo recetor recebesse e validasse a trama de dados enviava para o dispositivo emissor a trama de *acknowledgement*.

3 - Se o dispositivo emissor não recebesse a trama de *acknowledgement* dentro de um tempo válido, este voltava ao *TX\_Mode* e, se estivesse configurado para tal, retransmitiria automaticamente a trama (*auto retransmission*), caso contrário considerava o envio falhado.

Então em 3 casos distintos ocorreria uma interrupção no pino IRQ, que se traduzia por uma mudança lógica de 1 para 0, eram eles:

Envio sucedido - Numa comunicação com auto *acknowledgement* traduzia-se no envio de uma trama de dados e receção do respetivo *acknowledgement* numa janela de tempo válida. Numa comunicação sem auto *acknowledgement* significava que os dados tinham sido postos no meio;

Receção de dados - Significava que o transceptor tinha recolhido dados válidos, com o seu endereço e que estes estavam disponíveis para leitura na memória FIFO de receção;

Envio máximo de retransmissões - Para que ocorresse esta interrupção era necessário que o chip estivesse configurado com auto *acknowledgement* e com retransmissão automática em caso de falha. Nestas condições uma interrupção deste tipo significava que o dispositivo tinha reenviado os dados o número máximo de vezes para o qual estava configurado e que em todos esses reenvios não tinha recebido nenhum *acknowledgement*, como tal considerava o envio falhado.

Estes mecanismos descrevem, resumidamente, o funcionamento do nRF24L01+ conforme o protocolo Enhanced ShockBurst™. Da perspetiva do programador a implementação deste protocolo tornava-se simples uma vez que este apenas necessitava de configurar os registos associados através da interface SPI, interface a partir da qual também se poderiam escrever os dados a enviar e ler os dados recebidos. Assim sendo o programador poderia seleccionar se pretendia auto reconhecimento dos dados recebidos (auto *acknowledgement*), retransmissão de dados em caso de falha (auto *retransmission*), deteção de erros (CRC16 ou CRC8) e as interrupções que queria ver refletidas no pino IRQ, tudo isto de forma a tornar a comunicação mais eficiente e mais fidedigna.

#### 4.1.5 Plataforma central

A plataforma central foi desenvolvida com o intuito de dar ao utilizador um meio de monitorização e controlo dos sistemas da habitação sem que este tivesse necessidade de conhecer, de forma especializada, o *hardware*. Na definição da plataforma computacional foi adotada uma arquitetura centralizada de forma a se estabelecer um ponto central de controlo e monitorização de todo o sistema. As vantagens que levaram a esta escolha passavam pela maior facilidade na gestão e identificação de problemas da rede, uma vez que todos os dados poderiam ser consultados num único ponto, e na existência de maior segurança, dado que apenas seria necessário controlar um único ponto de acesso. Paralelamente, esta arquitetura permitia que, tanto na rede sem fios, como no barramento RS-485, fossem facilmente adicionados novos dispositivos e novas funcionalidades. Como desvantagens verificava-se a necessidade de maior processamento<sup>8</sup> e o facto de a arquitetura estar dependente do correto funcionamento da plataforma central.

Assim sendo e de forma a potenciar a arquitetura, minimizando o impacto das desvantagens referidas, seria fundamental o uso de uma poderosa ferramenta de desenvolvimento. Do reportório de conhecimentos de ferramentas, para desenvolvimento de *software* de interface gráfica, por parte do autor, algo que teve de ser levado em conta uma vez que à formação/especialização também estariam associados custos, evidenciavam-se duas ferramentas, designadamente o Visual Basic do pacote Microsoft Visual Studio e o Movicon, um *software* SCADA da Progea. Para seleccionar a ferramenta que melhor se adaptaria ao desenvolvimento do sistema, primeiro foi feita uma especificação dos requisitos com base no estudo da oferta das soluções comerciais existentes e nas necessidades da arquitetura proposta. Esses requisitos foram divididos em duas categorias, de acordo com o seu grau de relevância, designadamente em importantes e vantajosos. Os requisitos os quais eram considerados importantes, na elaboração desta dissertação, eram a implementação da comunicação MODBUS, o controlo e a monitorização dos dispositivos na rede, a compatibilidade com bases de dados, o acesso condicionado ao sistema, a gestão de utilizadores, a calendarização de tarefas, a gestão de alarmes e a videovigilância. Por outro lado os requisitos considerados vantajosos eram a facilidade de desenvolvimento, a compatibilidade com outros protocolos, o suporte de relatórios estatísticos (consumos energéticos, alarmes etc.), de serviços WEB, de controlo por voz e a configuração/diagnóstico da rede.

---

<sup>8</sup> Os módulos baseados em microcontroladores permitiam operações locais de controlo reduzindo em parte a necessidade de processamento.

Embora ambas as ferramentas permitissem criar uma plataforma que cumprisse os requisitos enunciados e ainda oferecessem algumas outras possibilidades, a escolha recaiu sobre o *software* SCADA Movicon. Do ponto de vista do autor, o Movicon permitia uma maior facilidade no desenvolvimento, apresentava uma maior compatibilidade com diversos dispositivos de automação/controlo e tinha uma maior aceitação no mercado de domótica, além de ainda disponibilizar uma boa documentação e ferramentas eficientes de gestão energética. Pese embora os custos associados a este *software*, a adoção de uma ferramenta na qual, já se orientavam esforços no desenvolvimento de produtos para o mercado de domótica (Movicon Building Automation Edition) tornava-se numa clara mais-valia na potencialização do sistema a desenvolver.

No anexo 1, é apresentada um orçamento para alguns dos diferentes tipos de licenças fornecidas para o Movicon 11.4<sup>9</sup>, no qual também podem ser consultadas algumas das potencialidades oferecidas pelo *software*. É também importante referir, que por uma questão de disponibilidade e conveniência, a plataforma foi desenvolvida num computador pessoal, mas poderia ser baseada num painel tátil, que suportasse o sistema operativo Windows CE, de forma a se adaptar melhor às exigências do mercado de domótica.

---

<sup>9</sup> A versão Movicon Building Automation é baseada no Movicon 11.4, na perspectiva do autor os preços apresentados devem ser equivalentes para as duas versões.

## 5. Implementação da solução proposta

No presente capítulo pretende-se descrever ao leitor a implementação da solução apresentada anteriormente. Como descrito o sistema baseava-se numa plataforma central e em módulos adicionais, partindo dessa organização segue-se a apresentação de cada subsistema.

### 5.1 Módulo Central

O módulo central era único na rede e era composto por um microcontrolador PIC18F26J50 com uma ligação série a um computador pessoal com a plataforma Movicon Building Automation, por uma placa de desenvolvimento MOD-RS485 e por um transceptor de radiofrequência nRF24L01+. A plataforma central permitia a interface com o utilizador, o transceptor RS-485 transformava os níveis lógicos de 3.3V em níveis diferenciais RS-485 e permitia fazer a interligação com o barramento e o transceptor RF permitia a comunicação com os módulos RF desenvolvidos. O esquema elétrico do módulo central pode ser consultado no anexo 2 e a sua constituição está apresentada na Figura 5.1.

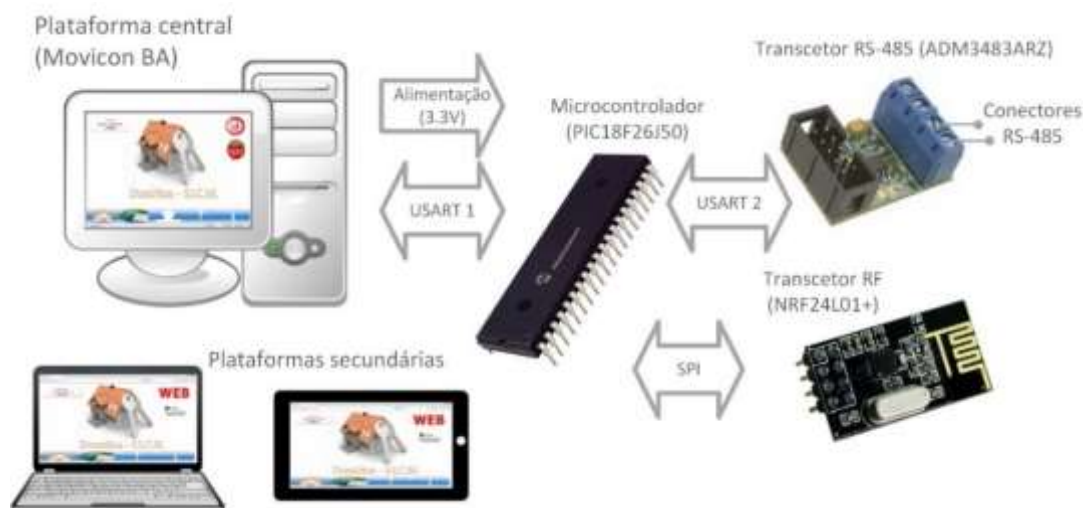


Figura 5.1 - Módulo central do sistema desenvolvido.

#### 5.1.1 Funcionamento do microcontrolador central

O microcontrolador foi adicionado ao módulo central de forma a possibilitar a interface com os meios de transmissão. Como tal o microcontrolador era usado para determinar, a partir do endereço da mensagem, se a estação de destino estava no meio cablado ou na rede sem-fios e reencaminhar a mensagem através do *hardware* respetivo. Posteriormente este, de acordo com o tipo de mensagem, deveria receber a resposta pelo mesmo meio físico, e reencaminhá-la para o computador. O diagrama de funcionamento e de configuração do microcontrolador central está apresentado na Figura 5.2.

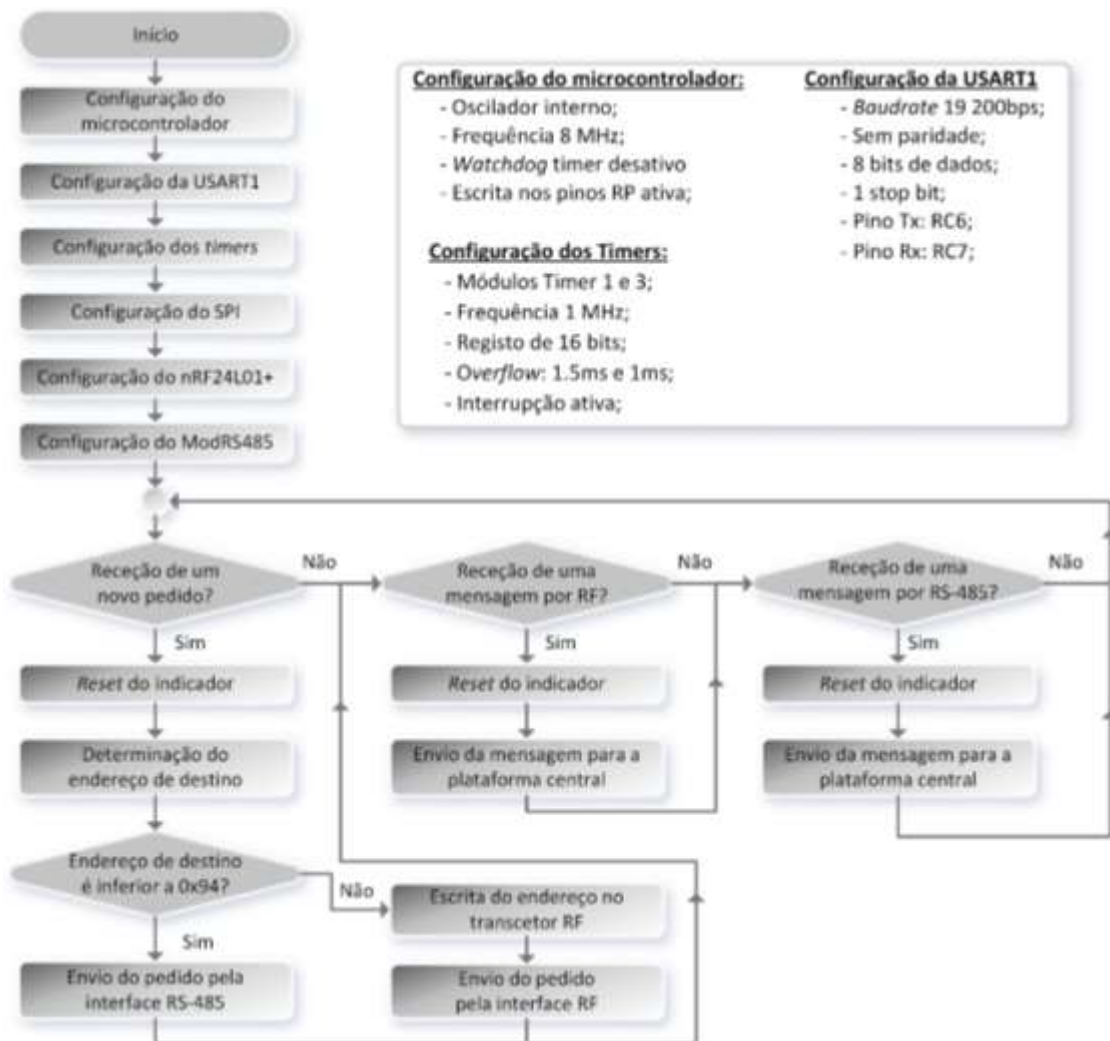


Figura 5.2 - Fluxograma do funcionamento do microcontrolador central.

Das configurações efetuadas no microcontrolador as configurações do SPI e do nRF24L01+ são iguais às efetuadas no módulo RF e serão descritas posteriormente, tal como a configuração do ModRS485 que é igual à usada no módulo RS-485. De resto salienta-se a configuração da porta série (módulo USART1) com a taxa de transmissão de 19 200bps, 8 bits de dados, 1 *stop* bit e sem paridade. Era através desta interface que eram recebidos/respondidos os novos pedidos da plataforma central. Podemos também observar, no canto inferior esquerdo da figura, a aplicação do esquema de endereçamento definido anteriormente.

Na imagem também é possível identificar que o funcionamento do ciclo principal do microcontrolador prendia-se nas condicionantes de existirem pedidos novos ou respostas aos pedidos efetuados. A leitura e subsequente indicação de novos pedidos/respostas era feita através de interrupções no sistema as quais funcionavam como apresentado no fluxograma na Figura 5.3.



Figura 5.3 - Fluxograma da rotina de interrupção do microcontrolador central.

Do fluxograma da figura os *timers* 1 e 3 eram usados para indicar a receção de uma nova mensagem pelos módulos USART, uma vez que nas tramas MODBUS RTU não existiam caracteres terminadores. O valor de *overflow* do *timer* 1 sinalizava a receção de um novo pedido da plataforma central (USART1) e foi definido com o valor de 1.5ms de forma a integrar o tempo máximo de espera entre bytes MODBUS. O tempo máximo de espera entre bytes era traduzido pela expressão:

$$T_{EsperaEntreBytes} \leq 1.5 \times T_{ByteMODBUS} \quad (3)$$

Assim sendo à taxa de transmissão definida de 19 200bps o tempo de *overflow* do *timer* deveria ser superior a  $1.5 \times 0.6\text{ms} = 0.9\text{ms}$ . Por sua vez o *timer* 3 sinalizava as respostas recebidas através do barramento (USART2). Este, além do tempo máximo de espera entre bytes, deveria também integrar o tempo de comutação do transceptor RS-485, mas como a taxa de transmissão definida era de 57 600bps o valor de *overflow* definido foi de 1ms.

Podemos também observar que a leitura dos dados do transceptor nRF24L01+ era feita na rotina de interrupção, isto deveu-se sobretudo à necessidade dos dados serem lidos assim que eram recebidos, uma vez que o atraso na leitura de uma sequência de dados poderia resultar numa sobreposição com dados novos. A leitura dos dados era feita através da interface SPI e o procedimento de leitura consistia na identificação da quantidade de dados presentes na FIFO, através da leitura do registo<sup>10</sup> R\_RX\_PL\_WID (*Read Received Payload Width*), e subsequente leitura dos dados da FIFO. É de referir que se o registo indicasse um valor superior a 32, como descrito anteriormente a FIFO tinha um tamanho máximo de 32 bytes, os dados eram inválidos, enquanto os restantes valores indicavam o tamanho da mensagem MODBUS e eram usados para reencaminhar a mensagem para a plataforma central.

<sup>10</sup> A descrição dos registos do nRF24L01+ pode ser consultada no capítulo 9 do manual do fabricante.

### 5.1.2 Desenvolvimento da plataforma Movicon

Na plataforma Movicon era feito o processamento dos dados da rede e a interface dos mesmos com o utilizador. O correto funcionamento da plataforma central era imprescindível no sistema desenvolvido, uma vez que era através desta que era possível a interligação dos subsistemas. Nesta secção descreve-se de uma maneira resumida o procedimento de adição e de configuração de um *driver* de comunicação e da adição de variáveis MODBUS, uma vez que esta era a base na qual estava assente toda a infraestrutura criada. O procedimento para adicionar o *driver* é o apresentado na Figura 5.4.

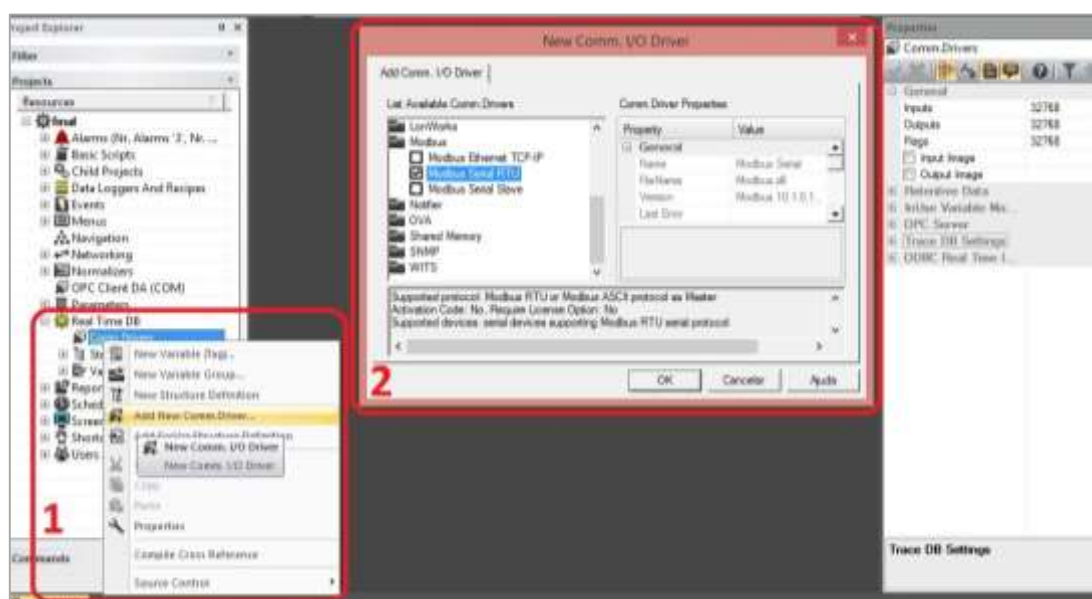


Figura 5.4 - Procedimento de adição do *driver* MODBUS no Movicon.

Depois de adicionado o *driver* dever-se-ia adicionar as estações que estivessem fisicamente conectadas no sistema. Para cada módulo devia-se adicionar uma estação na qual o *Station ID* correspondia ao endereço MODBUS do módulo, tal como apresentado na Figura 5.5.

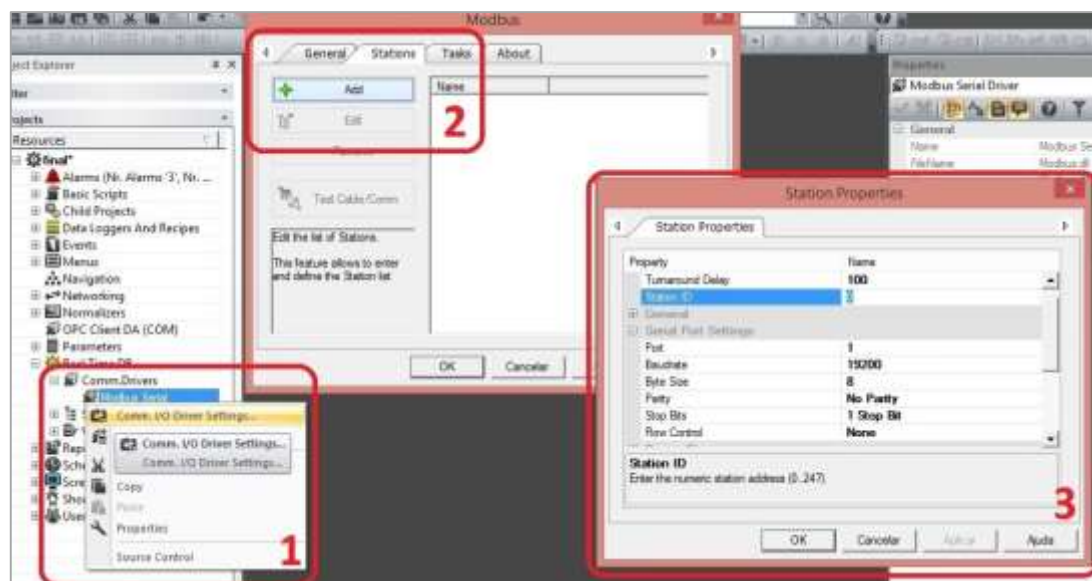


Figura 5.5 - Procedimento de adição de uma estação MODBUS no Movicon.



A configuração da porta série deveria corresponder exatamente à configuração efetuada na UART1 do microcontrolador central. Como tal a comunicação estabelecida teria uma taxa de transmissão de 19 200bps, 8 bits de dados, 1 *stop* bit e sem paridade. Posteriormente devia-se associar variáveis Movicon às variáveis MODBUS definidas nos módulos *slave* para que a variação física nos microcontroladores permitissem a demonstração dinâmica no Movicon. O procedimento para se associar variáveis MODBUS no Movicon é o apresentado na Figura 5.6.

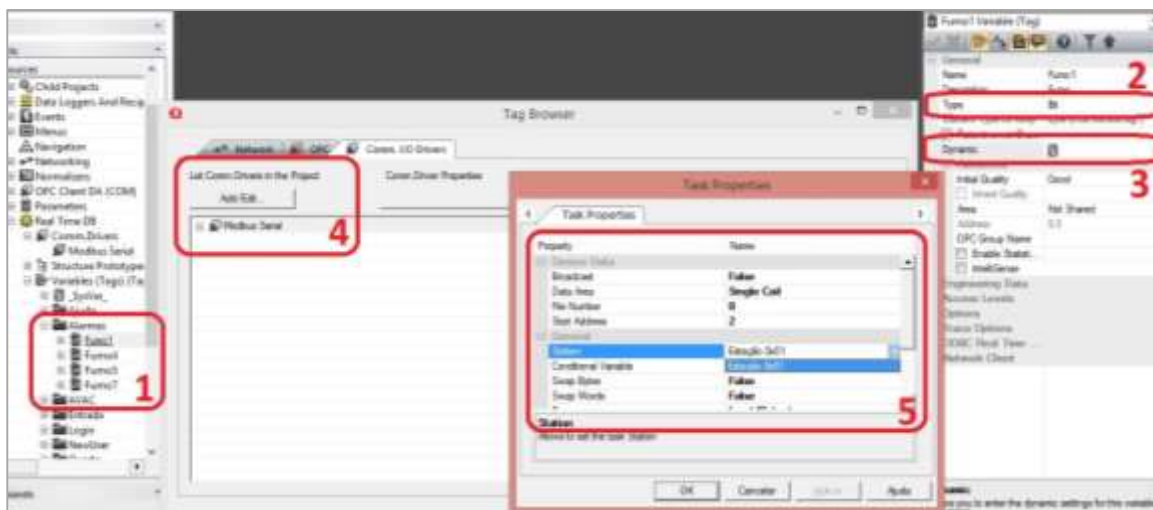


Figura 5.6 - Procedimento de definição de uma variável MODBUS no Movicon.

Na associação de variáveis devia-se ter atenção para que o tipo de dados definido no Movicon correspondesse ao estabelecido pelo MODBUS. Assim sendo e conforme implementado a associação das variáveis MODBUS com as variáveis Movicon e a tradução das variáveis físicas dos microcontroladores *slaves* foi feita de acordo com a Tabela 5.1.

Tabela 5.1 - Associação de variáveis MODBUS-Movicon-Microcontrolador.

Variável MODBUS	Função	Variável Movicon	Variável Microcontrolador	Endereços
<i>Single Coil</i>	01 Ler 05 Escrever	Bit	Saída digital	0 - 65535
<i>Discrete Input</i>	02 Ler	Bit	Entrada digital	0 - 65535
<i>Input Register</i>	04 Ler	<i>Word</i> (16 bits)	<i>Unsigned Int</i>	0 - 65535
<i>Single Register</i>	03 Ler 06 Escrever	<i>Word</i> (16 bits)	<i>Unsigned Int</i>	0 - 65535

Depois da construção da estrutura base, assente na associação de variáveis, procedeu-se à criação do protótipo da interface com o utilizador, que é apresentada de seguida.

### 5.1.3 Protótipo da interface com o utilizador

O protótipo de interface com o utilizador foi construído de forma a explorar as potencialidades do Movicon. A nível gráfico a interface baseava-se em ecrãs, nos quais estavam

dispostos serviços e informações que possibilitavam a autoaprendizagem no uso do sistema, por parte do utilizador, de forma simples e intuitiva. É de referir que a interação com os dispositivos não requeria qualquer necessidade de conhecimento sobre o *hardware*.

Um dos principais requisitos era a gestão de acesso dos utilizadores ao sistema. Assim sendo, o acesso à plataforma central era condicionado mediante a autenticação através de credenciais próprias. O ecrã de autenticação está apresentado na Figura 5.7.



Figura 5.7 - Ecrã de autenticação do utilizador.

Como estavam disponíveis diferentes tipos de serviços, alguns dos quais requeriam um certo nível de especialização, foram então associados diferentes níveis de permissões aos diferentes utilizadores. Um desses serviços era o de diagnóstico e de configuração da rede, o qual apenas estaria acessível ao administrador e ao instalador. Na Figura 5.8 está apresentado o ecrã desenvolvido para a gestão da rede.



Figura 5.8 - Ecrã de gestão da rede.

Outro pressuposto que de acordo com a perspectiva do autor faria claro sentido, por se tratar de um sistema residencial, era a existência de múltiplos utilizadores. Como tal, para gerir os utilizadores do sistema, após autenticação válida, seria possível adicionar ou remover utilizadores. Para se adicionar um utilizador apenas deveriam ser definidos os seus dados de autenticação, enquanto para remover um utilizador seria necessário, não sendo o administrador, saber os dados de autenticação do utilizador a remover. Estas validações foram criadas com recurso a um *script* de VBA. O ecrã de gestão de utilizadores está na Figura 5.9.

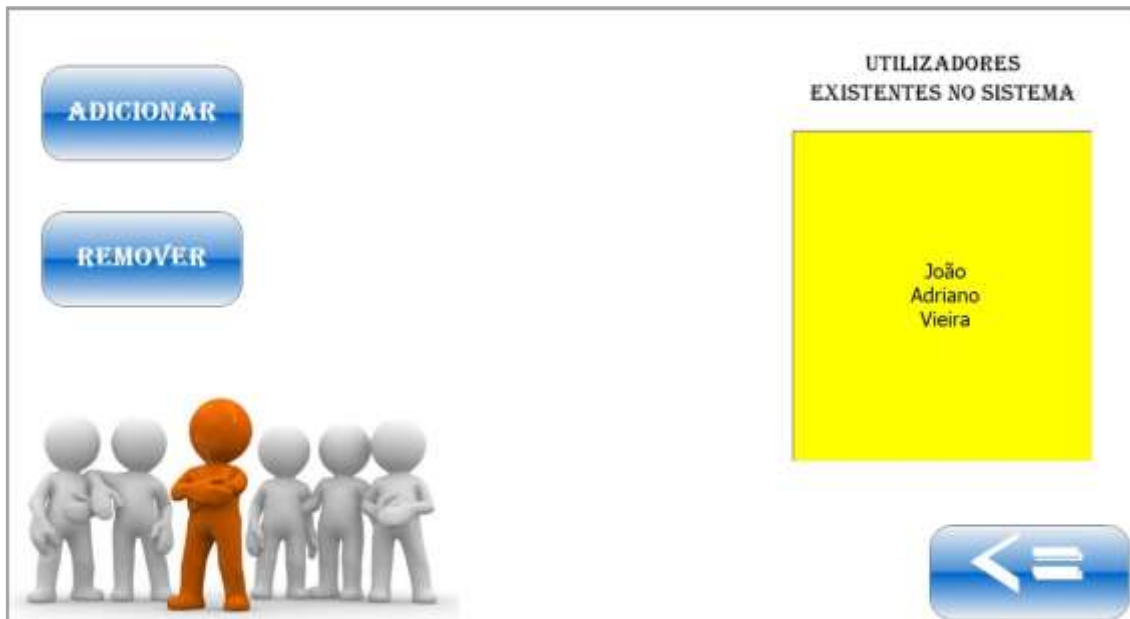


Figura 5.9 - Ecrã de gestão dos utilizadores.

A opção de gestão de utilizadores é uma das 5 que poderiam ser encontradas no menu principal. O ecrã principal era apresentado ao utilizador logo após a autenticação, e apresentava todas opções do sistema. A Figura 5.10 ilustra aquele que é o ecrã central do sistema.



Figura 5.10 - Ecrã principal.

As opções acessíveis através dos botões que estão presentes na parte inferior do ecrã eram: 1 - Monitorização e controlo da habitação; 2 - Serviços orientados para utilizadores WEB; 3 - Gestão de utilizadores; 4 - Ecrã de ajuda e de descrição da utilização do sistema; 5 - Ecrã de informação sobre o sistema e sobre o autor. Além destas opções também era apresentado o utilizador conectado no sistema, sendo possível a este sair ou encerrar o sistema, e no caso, como estava conectado o administrador, a consulta do ecrã de gestão da rede.

Assim sendo, para iniciar a utilização do sistema, era aconselhável ao utilizador consultar o ecrã de ajuda, onde seria possível encontrar informação sobre cada um dos menus e submenus e a descrição de como interagir com os mesmos. O ecrã de ajuda está na Figura 5.11.

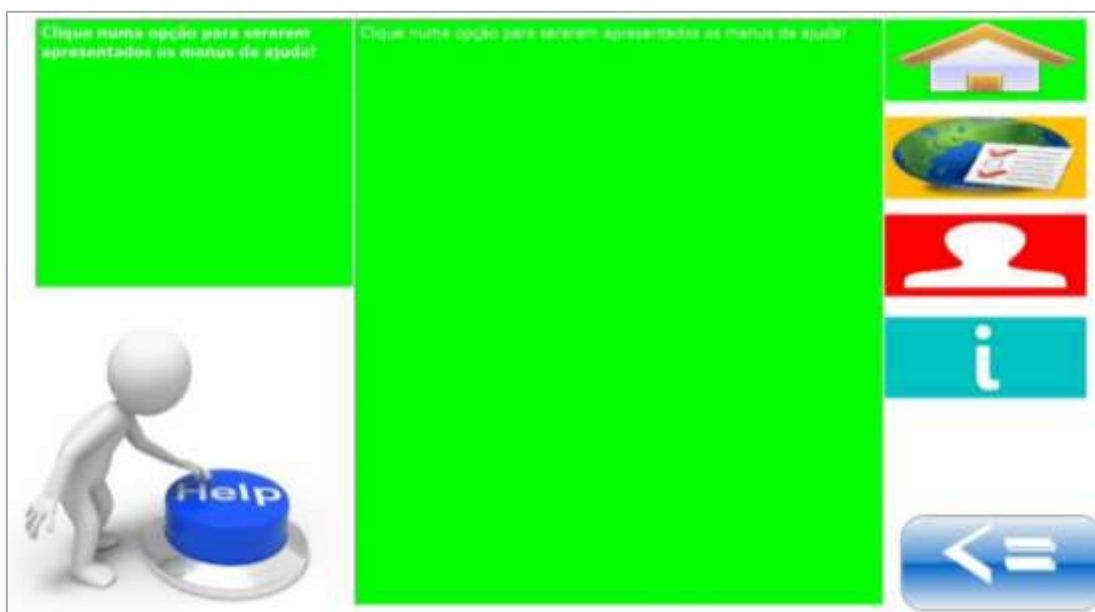


Figura 5.11 - Ecrã de ajuda.

Como podemos observar na figura temos os marcadores à direita que correspondem aos diversos ecrãs do sistema, estes ao serem clicados apresentam na caixa mais à esquerda os submenus e na caixa central as funcionalidades dos mesmos. Após a compreensão do método de interação com o sistema o utilizador poderia tirar total partido das funcionalidades propostas.

Partindo então do menu da casa, o utilizador poderia encontrar submenus com hiperligações para os diferentes ecrãs que davam interface aos diferentes serviços, através dos quais o utilizador poderia monitorizar e controlar a sua habitação. Na Figura 5.12 está apresentado o ecrã de submenus da casa, no qual podemos observar os sistemas a controlar listados à direita.

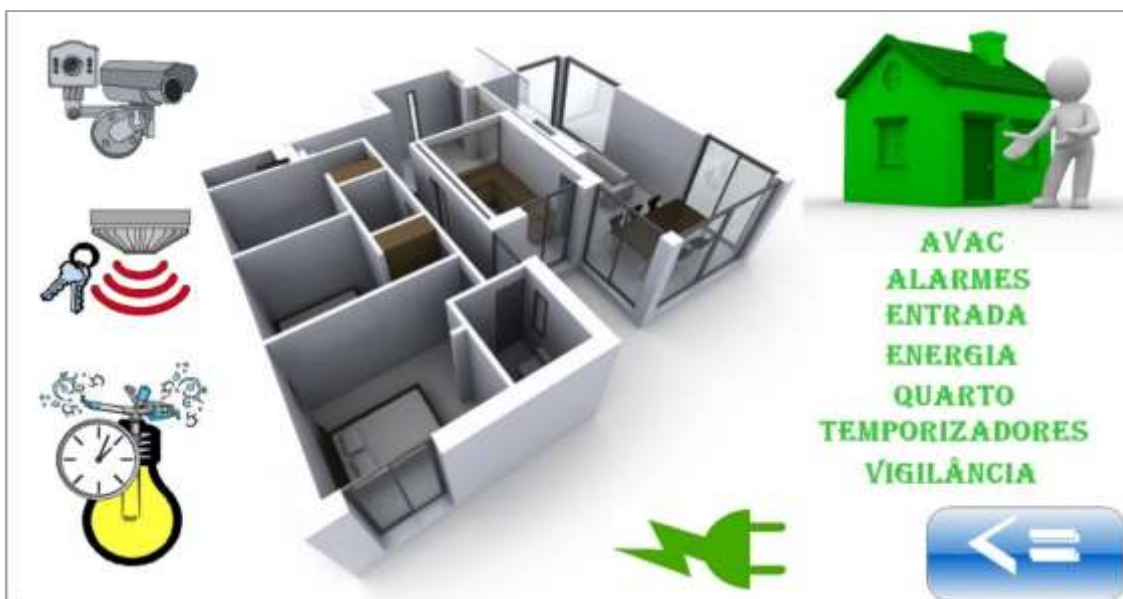


Figura 5.12 - Ecrã de submenus da casa.

O utilizador podia então usar a lista ou os ícones como hiperligação para os diferentes ecrãs do sistema. Dos serviços apresentados na figura, a maioria deles eram indispensáveis num sistema de domótica, outros, tais como a entrada e o quarto, foram exemplos usados de forma a validar, através de animações dinâmicas, a interação com os módulos convencionais desenvolvidos. Podemos então observar que o primeiro item dava acesso aos sistemas de AVAC. O ecrã que permitiria climatizar as diversas zonas da habitação está na Figura 5.13.

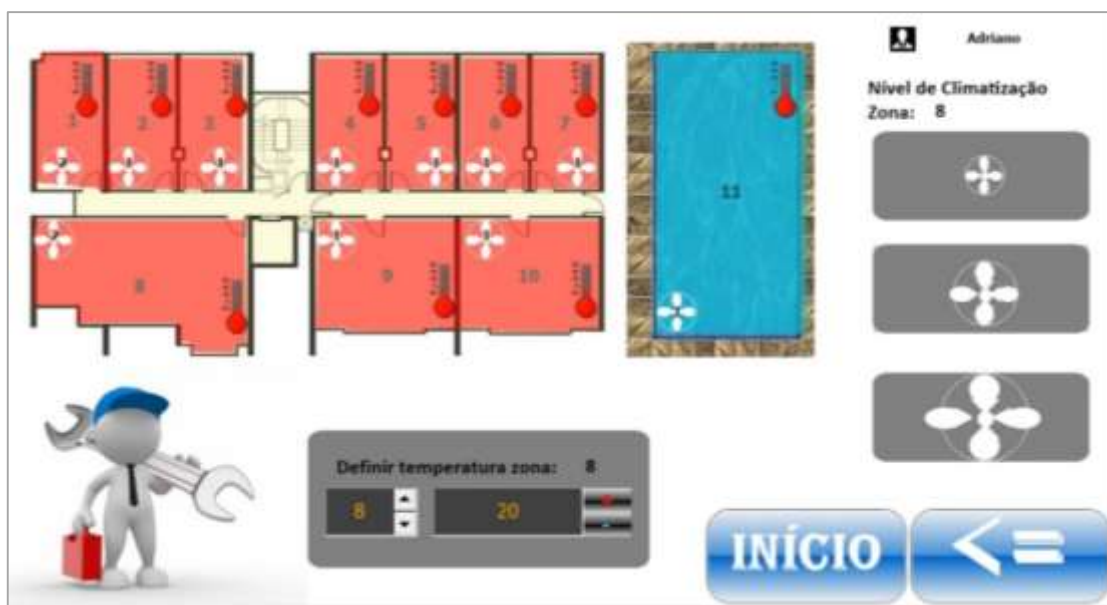


Figura 5.13 - Ecrã de controlo dos sistemas de AVAC.

Podemos observar na figura que as diferentes divisões estão numeradas por zonas, que funcionam como hiperligações através quais o utilizador poderia definir os *setpoints* de temperatura e de nível de climatização para as diferentes divisões.

O segundo item do menu casa permitia aceder aos Alarmes. O ecrã desenvolvido para monitorização dos alarmes está apresentado na Figura 5.14.



Figura 5.14 - Ecrã de monitorização dos alarmes.

Neste ecrã, teve-se especial atenção às exigências da monitorização de alarmes, uma vez que numa habitação, estes tem um cariz de especial importância, visto que permitem a deteção de ocorrências que podem pôr em causa a integridade da habitação ou dos habitantes. Desta forma teve-se o cuidado de configurar as variáveis de alarme com lógica negativa, ou seja o estado inativo corresponde a um valor lógico 1, assim contornava-se a não deteção de um alarme por inatividade do sensor. Por outro lado, sempre que ocorresse um alarme seria interrompida qualquer ação que estivesse a ser executada pelo utilizador na plataforma central e seria apresentado o ecrã “Alarmes” e o respetivo alarme ativo. Finalmente e de forma a facilitar a identificação da causa do alarme estes estavam divididos por zonas e por grau de gravidade. Foi também criada uma ferramenta para consultar relatórios com as estatísticas das diferentes ocorrências.

Como referido anteriormente alguns dos ecrãs aqui apresentados foram apenas criados de forma a validar a infraestrutura proposta. Um desses ecrãs funcionava como uma interface para uma Entrada e pretendia simular o ambiente no qual se adequava a implementação de um módulo RF. O ecrã Entrada está apresentado na Figura 5.15.





Figura 5.15 - Ecrã da entrada.

Neste ecrã apresentavam-se 4 animações distintas, nomeadamente a abertura/fecho do portão e da garagem, o ligar/desligar da iluminação exterior e o ligar/desligar do sistema de rega. Estes 4 diferentes meios de interação poderiam ser efetuados na plataforma central através dos botões no canto inferior esquerdo da figura ou então no módulo RF através do comando IR. Os sistemas de rega e de iluminação exterior poderiam também ser programados para se ligarem/desligarem num determinado horário em função do dia da semana. O ecrã que permitia a programação da rega e da iluminação exterior juntamente com os sistemas de aquecimento está apresentado na Figura 5.16.

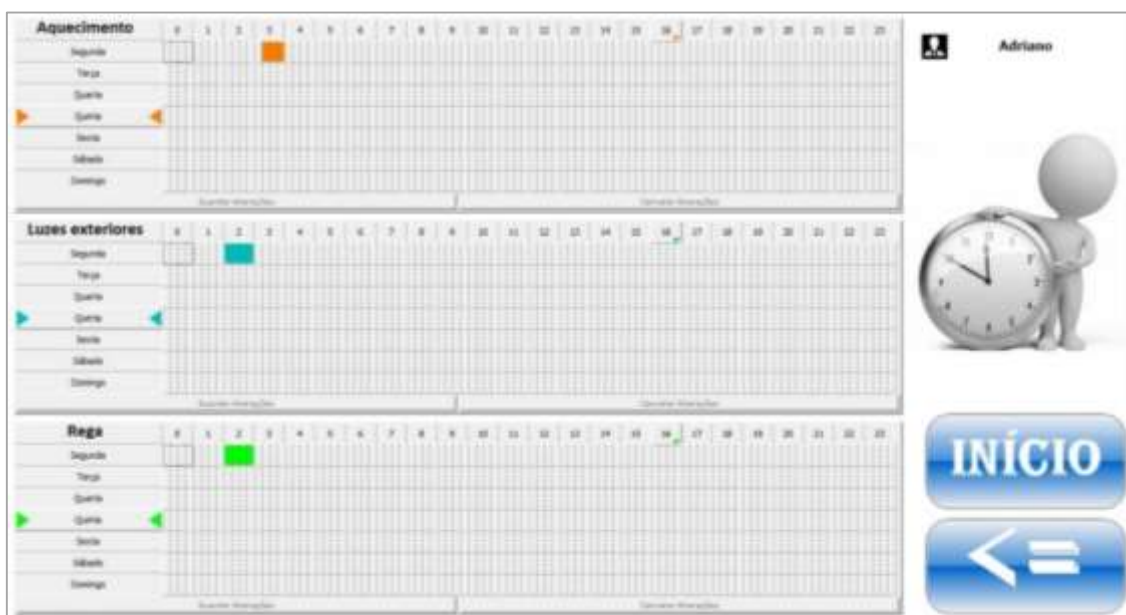


Figura 5.16 - Ecrã dos temporizadores.

De forma a fornecer ao utilizador uma perspetiva dos consumos energéticos da habitação foi também adicionado ao sistema um dispositivo que permitia ler variáveis energéticas. Para apresentar essas variáveis foi desenvolvido o ecrã apresentado na Figura 5.17.



Figura 5.17 - Ecrã de medição de energia.

Podemos observar na imagem que eram apresentadas ao utilizador, através de caixas de texto, as variáveis elétricas com cariz mais importante, no gráfico era apresentada a potência ativa ao longo do tempo e nos contadores eram guardado os consumos de energia em kVARh e em kWh. Era também dada ao utilizador a opção de consultar relatórios de consumos em função de períodos específicos. O módulo que permitia ler estes valores é apresentado na secção 5.2.6 (pg. 74). Na Figura 5.18 está apresentado o ecrã que permitiria ao utilizador efetuar a vídeo vigilância da sua habitação.



Figura 5.18 - Ecrã de videovigilância.



O Movicon dá suporte ao uso de câmaras IP, de maneira a fornecer ao utilizador serviços de vigilância da habitação, foi desenvolvido o ecrã da Figura 5.18, mas como o autor não tinha acesso a câmaras IP, de forma a validar o ecrã, foram usados URLs de imagens alocadas *online*.

Por último e de forma a dar suporte a serviços WEB foi criado um ecrã no qual o utilizador da plataforma central poderia publicar tarefas, as quais apareceriam ao utilizador que acesse remotamente ao sistema através de um navegador WEB. O ecrã de tarefas WEB criado está apresentado na Figura 5.19.

	Hora de Adição	Tarefa	Utilizador	Prazo Final	Prioridade	Estado
1	2014/05/23 16:46:55	Compro	Manuel	15:30	4	Realizada
2	23/05/2014 17:22:18	Limpar	Adriano	17:00	1	Realizada
3	23/05/2014 22:54:00	Jeitar	Mano	20:00	2	Por Realizar

qui 29 mai  
 Adriano  
 17:24:45

Tarefa
Prazo Final
Prioridade

ADICIONAR TAREFA
REMOVER TAREFA
ATUALIZAR

Tarefa Nr.

Figura 5.19 - Ecrã de tarefas WEB.

Neste ecrã as tarefas e as suas propriedades são definidas através dos botões e caixas de texto na parte inferior do ecrã. As tarefas estão organizadas conforme a hora de adição, e conforme a tarefa publicada está associada, um prazo, uma prioridade e o seu estado atual (realizada ou por realizar). Por sua vez o utilizador WEB apenas pode consultar as tarefas e marcá-las como realizadas. Todas estas validações foram feitas com recurso a um *script* VBA.

## 5.2 Módulos externos

Os módulos externos eram aqueles que poderiam ser adicionados ao sistema, quer fossem módulos convencionais ou dispositivos específicos. Nesta subsecção pretende-se descrever ao leitor os módulos externos adicionados e o comportamento destes na rede.

### 5.2.1 Módulos RS-485

Os módulos RS-485 foram concebidos de forma a serem interligados com o meio cablado e possibilitavam o controlo descentralizado dos dispositivos ligados diretamente a estes. A interação com estes módulos poderia ser feita através da plataforma central ou através do

comando IR e a mudança local do estado dos dispositivos ligados implicava a sua atualização na plataforma central.

Estes módulos eram compostos por um microcontrolador PIC18F26J50, por uma placa de desenvolvimento MODRS485, um recetor IR TSOP34136 e alimentação externa de 3.3V. O microcontrolador possibilitava a ligação de entradas digitais e analógicas, saídas digitais e a escrita/leitura de registos de memória de 2 bytes, o transceptor RS-485 transforma os níveis lógicos de 3.3V em níveis diferenciais RS-485 e permitia fazer a interligação com o barramento e o recetor IR permitia ao utilizador interagir com o módulo através de um comando IR. O esquema elétrico do módulo RS-485 pode ser consultado no anexo 3 e a sua constituição está apresentada na Figura 5.20.

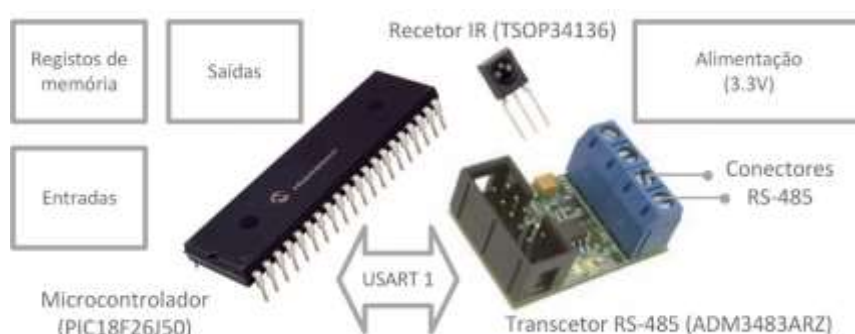


Figura 5.20 - Módulo RS-485 do sistema desenvolvido.

A atualização dos dispositivos ligados a este módulo era efetuada periodicamente através de pedidos MODBUS enviados pela plataforma central. O funcionamento e a configuração dos módulos RS-485 estão apresentados na Figura 5.21.

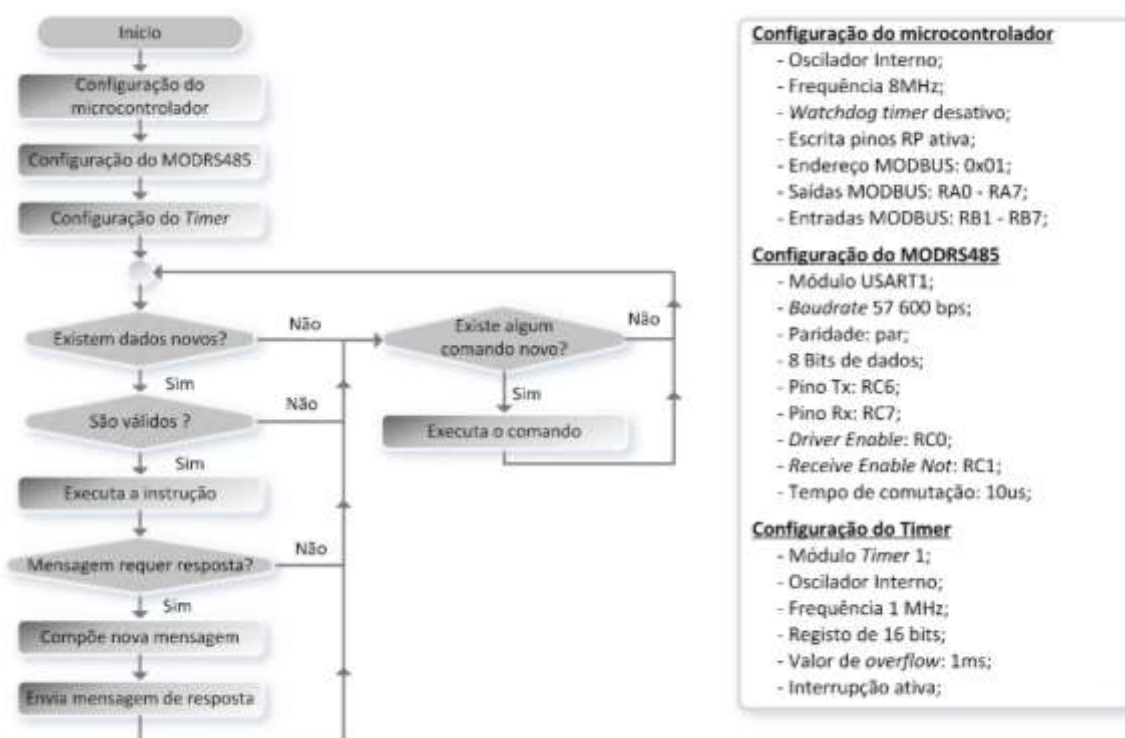


Figura 5.21 - Fluxograma do funcionamento do Módulo RS-485.

Podemos observar na Figura 5.21 que, o funcionamento do ciclo principal destes módulos, estava condicionado ao facto de existirem novos pedidos ou comandos executados pelo utilizador através do comando IR. A leitura e subsequente indicação de novos pedidos, ou comandos, eram feitas através de interrupções no sistema as quais funcionavam como apresentado no fluxograma da Figura 5.22.

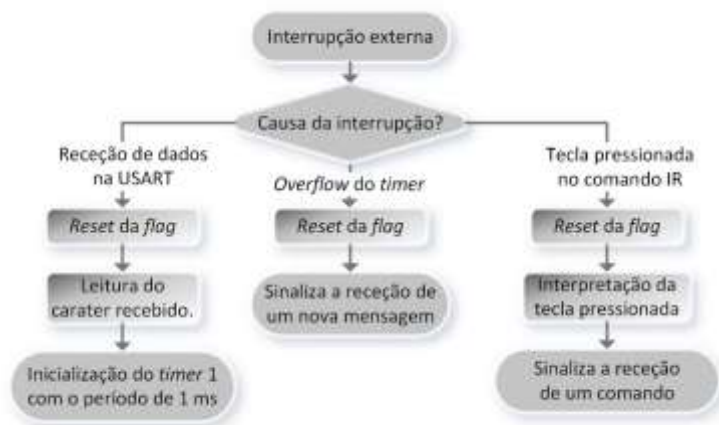


Figura 5.22 - Fluxograma da rotina de interrupção do Módulo RS-485.

Do fluxograma acima o procedimento de interpretação do comando IR é apresentado na secção 5.2.4 (pg. 72). O *timer* 1 foi usado para indicar a receção de uma nova mensagem completa, uma vez que nas tramas MODBUS RTU não existiam caracteres terminadores. O valor de *overflow* do *timer* 1, que sinalizava a receção de um novo pedido, foi definido com o valor de 1ms de forma a integrar e superiorizar o tempo de comutação do transceptor juntamente com o tempo máximo de espera entre bytes à taxa de 57 600bps.

### 5.2.2 Módulos RF

OS módulos RF foram concebidos de forma a serem interligados com o meio RF e possibilitavam o controlo descentralizado dos dispositivos ligados diretamente a estes. A interação com estes módulos poderia ser feita através da plataforma central, ou através do comando IR e a mudança local do estado dos dispositivos ligados, implicava a sua atualização na plataforma central.

Estes módulos eram compostos por um microcontrolador PIC18F26J50, um transceptor RF nRF24L01+, um recetor IR TSOP34136 e alimentação externa de 3.3V. O microcontrolador possibilitava a ligação de entradas digitais e analógicas, saídas digitais e a escrita/leitura de registos de memória de 2 bytes, o transceptor de radiofrequência permitia comunicar com o módulo central e o recetor IR permitia ao utilizador interagir com o sistema através de um comando IR. O esquema elétrico do módulo RF pode ser consultado no anexo 4 e a sua constituição está apresentada na Figura 5.23.

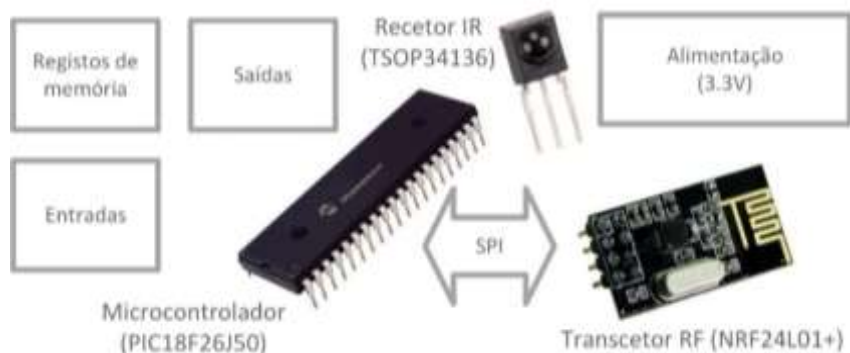


Figura 5.23 - Módulos RF do sistema desenvolvido.

A atualização dos dispositivos ligados a este módulo era efetuada periodicamente através de pedidos MODBUS enviados pela plataforma central. O funcionamento e a configuração dos módulos RF estão apresentados na Figura 5.24.

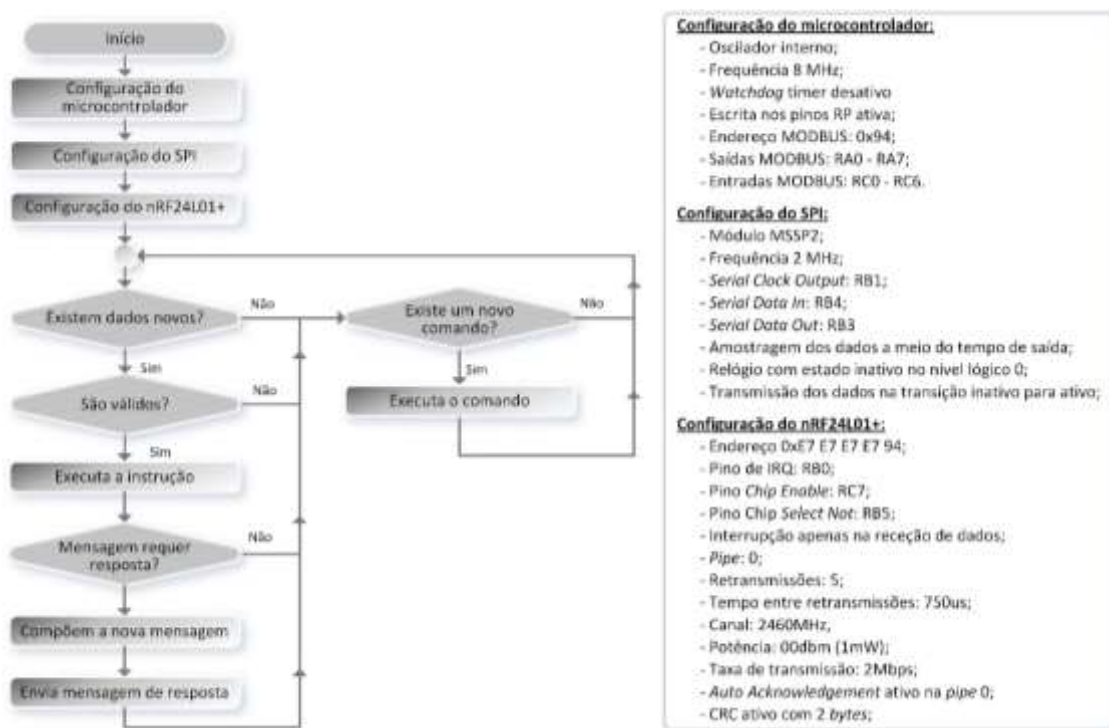


Figura 5.24 - Fluxograma do funcionamento do Módulo RF.

Da figura podemos observar que o transcetor RF foi configurado com uma taxa de transmissão de 2Mbps a uma potência de 1mW. Como descrito na secção 3.3 (pg. 40) a potência e a taxa de transmissão deveriam ser definidas em função da distância ao dispositivo *master* e da suscetibilidade a sinais provenientes de outros dispositivos RF. Para efeitos de conceção o configurou-se o transcetor com potência máxima (por ser único na rede) e à taxa de transmissão máxima por se localizar próximo do dispositivo *master*.

Tal como no Módulo RS-485, pode-se observar que o funcionamento do ciclo principal do Módulo RF estava condicionado ao facto de existirem dados novos ou comandos executados através do comando IR. A leitura e subsequente indicação de novos pedidos, ou comandos, eram

feitas através de interrupções no sistema as quais funcionavam como apresentado no fluxograma da Figura 5.25.



Figura 5.25 - Fluxograma da rotina de interrupção do Módulo RF.

No fluxograma acima, o procedimento de interpretação do comando IR é apresentado na secção 5.2.4 (pg. 72). A leitura dos dados do nRF24L01+ era também efetuado na rotina de interrupção e o procedimento de leitura era o mesmo que é descrito anteriormente para o microcontrolador central.

### 5.2.3 Implementação do protocolo MODBUS

A plataforma central efetuava periodicamente pedidos de informação através do envio de mensagens MODBUS para os módulos externos presentes no sistema. De forma a prover os módulos com ferramentas que permitissem executar as instruções pedidas e responder de forma correta aos pedidos efetuados o autor teve de desenvolver diversas rotinas as quais permitissem a configuração do módulo como um dispositivo MODBUS, a validação das mensagens recebidas, a execução dos pedidos recebidos e a respetiva resposta, se necessária, aos pedidos recebidos. As rotinas criadas que que permitiam que os módulos desenvolvidos funcionassem como dispositivos MODBUS foram:

*modbusConfigDevice()* - rotina onde era definido o endereço MODBUS do módulo, onde eram configuradas as saídas/entradas digitais e onde eram declarados os registos de memória de 2 bytes.

*modbusCalcCRC()* - rotina onde eram determinados os dois bytes de verificação de erros do CRC.

*modbusValidateMessage()* - esta rotina verificava a integridade da mensagem MODBUS recebida, através da rotina *modbusCalcCRC()*, e devolvia uma variável que indicava a sua validade/invalidade.

*modbusExecuteInstruction()* - esta rotina era executada após a validação do pedido recebido e consistia na tradução das funções MODBUS implementadas. Associada a esta função

estava uma variável que seria devolvida com o tamanho da nova mensagem, caso o pedido da plataforma central necessitasse de resposta. Essa variável, juntamente com a rotina *modbusCalcCRC()*, era posteriormente usada para compor a nova mensagem.

#### 5.2.4 Descodificação do comando IR

Para dar suporte à interação através de um comando de infra vermelhos foi adicionado a cada um dos módulos criados um recetor IR TSOP34136. Este recetor funcionava com uma alimentação de 3.3V e permitia receber sinais com frequências de 36kHz. O comando utilizado foi o de um transmissor FM e leitor de MP3. O comando e o recetor usados estão apresentados na Figura 5.26.



Figura 5.26 - Comando IR e recetor IR usados.

O comando seguia um protocolo específico de transmissão o qual não se encontrava muito documentado, pelo que a abordagem tomada para decodificar os diferentes botões foi através da captura do sinal, de todos os botões, com recurso ao picoscópio. Na Figura 5.27 está apresentado o formato da trama IR.

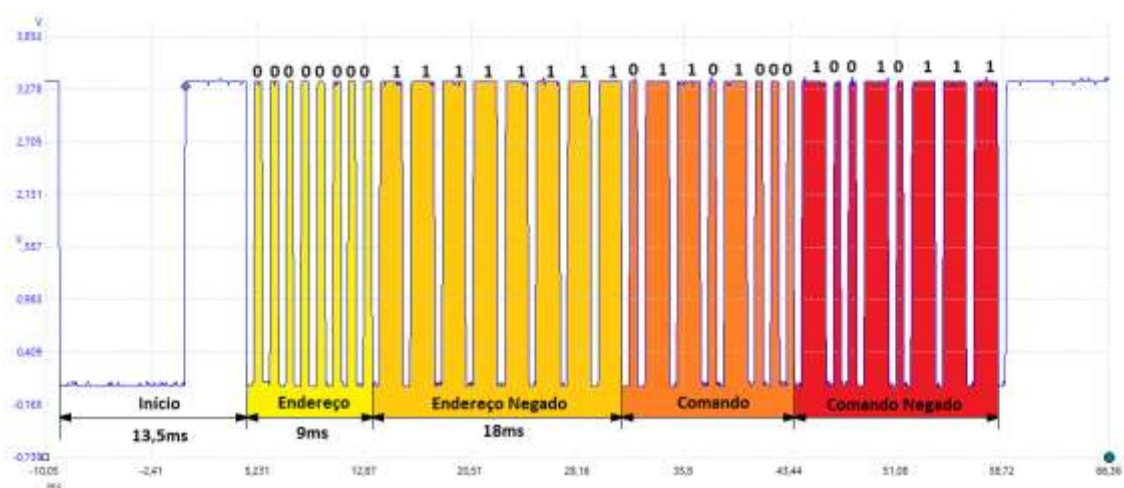


Figura 5.27 - Trama IR que codificava o botão 0 capturada com o picoscópio.

Podemos observar a partir da imagem que a trama recebida contém 4 campos de 8 bits, nomeadamente o endereço, o endereço negado, o comando e o comando negado. Para todos os botões o endereço e o endereço negado tinham o mesmo valor, respetivamente 0x00 e 0xFF.



Assim sendo para se determinar os botões apenas seria necessário interpretar o campo que correspondia ao comando. Como tal a interpretação baseou-se nos seguintes pressupostos:

1 - A interrupção devido à mudança lógica inicial de 1 para 0 no sinal indicava o início da trama, e correspondia ao pressionar de um botão.

2 - A leitura dos 8 bits correspondentes ao comando deveria ser efetuada  $13.5 + 9 + 18 = 40.5\text{ms}$  depois de a interrupção ocorrer.

3 - O valor lógico 0 correspondia a um espaço de  $562.5\mu\text{s}$  seguido de um pulso de  $562.5\mu\text{s}$ .

4 - O valor lógico 1 correspondia a um espaço de  $562.5\mu\text{s}$  seguido de um pulso de  $1.6875\text{ms}$ .

5 - O contínuo pressionar de qualquer tecla apenas continha a sequência de início e o restante sinal tinha valor lógico 1.

### 5.2.5 Placas protótipo

Com o intuito de desenvolver uma solução generalizada que pudesse ser usada como um módulo RF ou como um módulo RS-485 o autor concebeu um modelo para uma placa protótipo com recurso ao *software* CAD Eagle.

Esta placa incluía o microcontrolador PIC18F26J50, o recetor IR TSOP34136, incluía também níveis de tensão comumente usados em interfaces de controlo e relés (5V e 12V), diversos reguladores para diferentes tensões de entrada, que permitiam valores de alimentação de 12V, 5V, 5V através de USB ou 3.3V através da UEXT. Além da alimentação a ficha UEXT da OLIMEX também permitia conectar as placas MODRS485 com o transceptor ADM3483ARZ. A placa também continha as ligações redundantes que possibilitavam o seu uso como qualquer um dos dois módulos propostos. O esquema elétrico das placas protótipo pode ser consultado no anexo 5 e a sua constituição está apresentada na Figura 5.28.

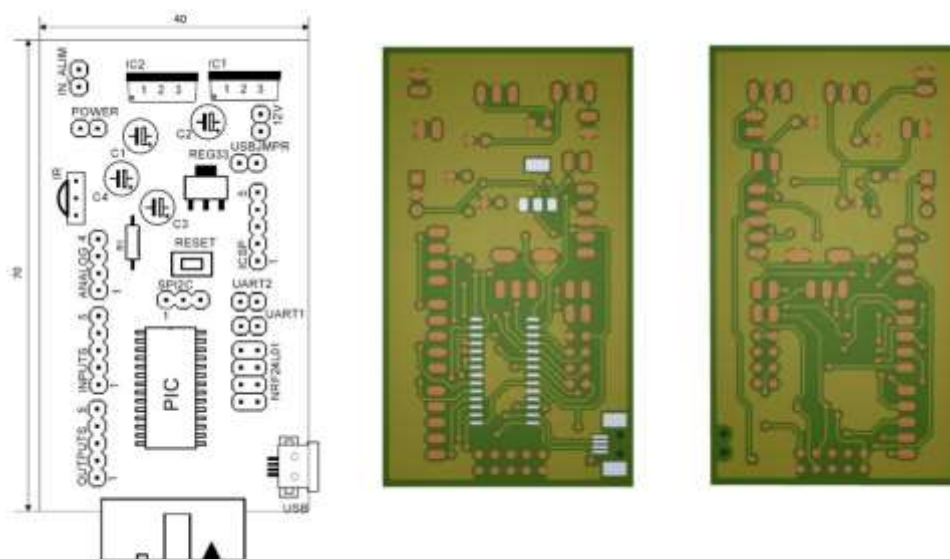


Figura 5.28 - Componentes e circuito impresso da placa protótipo.

Foi também efetuado um pedido de um orçamento junto de um representante da Seeed Technology<sup>11</sup> para a placa em questão o qual é apresentado no anexo 6.

### 5.2.6 Módulo de medição de energia

Como referido anteriormente poderiam ser adicionados outros tipos de dispositivos ao sistema desde que compatíveis com o barramento RS-485 e cujo protocolo de comunicação fosse suportado pelo Movicon. De forma a cumprir os objetivos propostos de aconselhamento energético foi adicionado ao sistema um dispositivo que permitiu efetuar medições de variáveis energéticas e permitiu também verificar a compatibilidade do barramento RS-485 com outros dispositivos. O dispositivo adicionado foi o WM22-DIN, apresentado na Figura 5.29.



Figura 5.29 - Dispositivo de medição de energia WM22-DIN [51].

Para se usar este dispositivo teve de se reconfigurar as velocidades de transmissão no barramento para 9600bps, uma vez que o formato da comunicação do dispositivo não era configurável. O dispositivo implementava as funções MODBUS 0x04 e 0x06 as quais correspondiam à leitura de registos de memória do tipo *input register* e à escrita de registos do tipo *single register*, respetivamente. Através destas funções era possível analisar a tensão da linha, a corrente, a potência ativa, reativa e aparente, o fator de potência, entre outras variáveis elétricas, isto tanto para sistemas trifásicos como monofásicos. Poderiam também ser lidos contadores que contabilizavam os consumos de energia em kWh e kVarh, e alarmes configurados para dispararem em função dos valores de diferentes variáveis. A este dispositivo poderiam também ser adicionados módulos com saídas digitais e analógicas que poderiam ser usadas para alarmes, para alimentação externa, etc.

Embora o dispositivo apresentasse bastantes potencialidades, este foi apenas usado para apresentar os consumos elétricos, através da medição das potências, da corrente, tensão e desfasamento na linha e de contadores globais, tal como apresentado anteriormente na Figura 5.17 (pg. 66).

---

<sup>11</sup> <http://www.seeedstudio.com/depot/>



## 6. Conclusão

Após a realização deste trabalho é possível afirmar que o objetivo proposto de criar um sistema de monitorização e controlo para os dispositivos e tecnologias presentes num ambiente residencial foi atingido com sucesso. Para atingir este objetivo foi proposta uma arquitetura centralizada, baseada no *software* SCADA Movicon, com subsistemas modulares baseados em microcontroladores interligados, ou num barramento RS-485, ou num meio RF, segundo o protocolo da camada de aplicação MODBUS.

Da realização do estudo inicial do estado da arte resultou a adoção dos módulos microcontrolados, que permitiam implementar localmente diversos subsistemas de I/O. Esta opção resulta numa mais fácil expansão do sistema, numa mais fácil manutenção da rede e numa redução da necessidade de processamento, na plataforma central, pela possibilidade de controlo local. Em contrapartida surgia a necessidade de se adotar um protocolo da camada de aplicação que definisse o método de atualização e de interação com os subsistemas. Como tal utilizou-se um protocolo de comunicação já definido e utilizado industrialmente, o MODBUS. Este revelou-se eficiente e fiável, e por ser um protocolo aberto com compatibilidade com diversos meios físicos de transmissão tornava a solução com um custo de implementação mais baixo e com uma maior aceitação no mercado.

Posteriormente, de forma a selecionar os meios físicos de transmissão a adotar, foi feita uma avaliação de alguns dos meios já definidos pelos protocolos de domótica, em paralelo com a definição dos requisitos da solução proposta. Foram então implementados dois meios de transmissão um cablado e um não cablado, designadamente par entrançado, de acordo com o protocolo RS-485, e RF com recurso aos transceptor nRF24L01+ e ao protocolo Enhanced ShockBurst™. A proposta dos dois meios distintos permitia adaptar o sistema às dificuldades da instalação, à distância entre os dispositivos e a plataforma central não pondo em causa os custos de implementação, o nível de segurança e a imunidade a ruídos e acessos indevidos.

Na criação da plataforma central e da interface com o utilizador o ambiente de desenvolvimento do Movicon permitia a elaboração de interfaces intuitivas, a fácil integração de diferentes sistemas de domótica de diferentes protocolos, entre eles BacNet, Konnex-EIB e Lonworks, apresentava compatibilidade com diversos sistemas de automação usados em sistema de AVAC, gestão energética e vídeo vigilância e funcionava como um servidor local, proporcionando o acesso ao sistema através de plataformas remotas. A nível de aconselhamento energético, que era um dos objetivos propostos, foram usados métodos fornecidos pelo Movicon para avaliar e classificar resultados, nomeadamente gráficos e relatórios detalhados que, com a integração de um dispositivo para monitorizar os consumos

energéticos, possibilitavam a melhoria da eficiência energética através de um meio de supervisão.

A nível de custos, concluiu-se que pese embora o facto de o sistema ter sido desenvolvido no âmbito de investigação, a infraestrutura criada dava algumas garantias de competitividade com as ofertas presentes no mercado de domótica, salientando-se que a oferta de compatibilidade com diferentes soluções de diferentes fabricantes tornou-se claramente numa mais-valia pois eliminava a necessidade de fidelização por parte dos clientes. Ressalva-se também o potencial tecnológico dos módulos convencionais concebidos.

Na gestão da rede, o MODBUS definia funções de diagnóstico o que paralelamente com o registo dos pedidos recebidos/respondidos permitiria avaliar a eficiência da comunicação e o estado dos dispositivos na rede. Adicionalmente a redefinição dos endereços dos dispositivos na rede, através da plataforma central, resultaria no enriquecimento do sistema, embora fosse criado um ecrã de suporte a estes serviços, estes não foram implementados devido à sua complexidade, e necessidade de se recorrer a *scripts* de VBA.

Por fim o estudo teórico e o trabalho prático, efetuado no tema proposto, culminaram num conjunto de novas ideias para trabalhos futuros, que da perspectiva do autor acrescentariam valor ao trabalho realizado. Neste âmbito seria interessante prover o sistema da capacidade de encaminhar mensagens através de redes que utilizem o protocolo internet, ou através da adição de um *gateway* ligado numa outra interface física na plataforma central ou através da implementação da *stack* IP no microcontrolador central. Seria também interessante melhorar a plataforma central de forma a potenciar os serviços fornecidos pelo Movicon, e adicionar um meio de configuração dos dispositivos na rede através da implementação de métodos que permitissem, entre outras coisas, definir o endereço dos dispositivos. A nível de aconselhamento energético apenas foram implementadas as medidas de monitorização e de exposição dos consumos, estas deveriam ser complementadas com a leitura de outras variáveis como a iluminação e a temperatura exterior, a humidade do solo ou a deteção de presença de forma a possibilitarem a criação de perfis energéticos para o controlo inteligente dos consumos em função das necessidades da habitação e dos contratos de energia.

## Bibliografia

- [1] **EuroX10 : X10Info** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:<http://www.eurox10.com/Content/x10information.htm>.
- [2] **SmartHome Discounts: Product Info** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:<http://www.smarthomediscounts.com/proddetail.aspx?id=18>.
- [3] **EuroX10 : Teoria da transmissão** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:<http://www.eurox10.com/Content/X10SignalTheory.htm>.
- [4] **What is KNX? - KNX Association [Official website]** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:<http://www.knx.org/knx-en/knx/association/what-is-knx/index.php>.
- [5] **Main advantages - KNX Association [Official website]** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:<http://www.knx.org/knx-en/knx/benefits-advantages/main-advantages/index.php>.
- [6] **KNX Topology - ComfortClick** - [Em linha] [Consult. 2 set. 2014]. Disponível em WWW:URL:[http://www2.komfortklik.si/mediawiki/index.php?title=KNX\\_Topology](http://www2.komfortklik.si/mediawiki/index.php?title=KNX_Topology).
- [7] KNX\_ASSOCIATION - KNX System Specifications Architecture. [s.d.]).
- [8] KNX\_ASSOCIATION - KNX System Specifications Interworking. [s.d.]).
- [9] **BACnet Overview** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:<http://www.bacnet.org/Overview/index.html>.
- [10] **BACnet: The Global Standard for Building Automation and Control Networks (Sustainable Energy): Michael H. Newman: 9781606502884: Amazon.com: Books** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:[http://www.amazon.com/BACnet-Standard-Building-Automation-Sustainable/dp/1606502883/ref=sr\\_1\\_1?ie=UTF8&qid=1408738007&sr=8-1&keywords=bacnet](http://www.amazon.com/BACnet-Standard-Building-Automation-Sustainable/dp/1606502883/ref=sr_1_1?ie=UTF8&qid=1408738007&sr=8-1&keywords=bacnet).
- [11] BACNET INTERNATIONAL - Introduction to BACnet. [s.d.]).
- [12] **BACnet Routers** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:<http://www.bacnet.org/Tutorial/HMN-Overview/sld028.htm>.
- [13] **AutomatedBuildings.com Article - The Protocol Implementation Conformance Statement (PIC)** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:<http://www.automatedbuildings.com/news/nov10/articles/opengeneral/101029125101opengeneral.html>.
- [14] NEWMAN, H. M. - BACnet Explained. **ASHRAE Journal**. 55:November (2013).
- [15] **The Language of BACnet** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:<http://www.bacnet.org/Bibliography/ES-7-96/ES-7-96.htm>.
- [16] ASHRAE ASSOCIATION - Proposed Addendum an to Standard 135-2012 , BACnet - A Data Communication Protocol for Building Automation and Control Networks. 2014).
- [17] **C-Bus Enabled** - [Em linha] [Consult. 3 set. 2014]. Disponível em WWW:URL:<http://www.cbus-enabled.com/what-cbus.htm>.
- [18] CLIPSAL AUSTRALIA - Serial Interface User Guide. December (2008a) 1–62.

- [19] CLIPSAL AUSTRALIA - C-Bus Application Messages & Behaviour. December (2008b).
- [20] **CEBus Demystified: The ANSI/EIA 600 User's Guide: Grayson Evans: 0639785328605: Amazon.com: Books** - [Em linha] [Consult. 4 set. 2014]. Disponível em WWW:URL:<http://www.amazon.com/CEBus-Demystified-ANSI-Users-Guide/dp/0071370064>.
- [21] **AV and Automation Industry eMagazine - INTRODUCTION TO THE CEBUS® COMMUNICATIONS PROTOCOL | HomeToys** - [Em linha] [Consult. 4 set. 2014]. Disponível em WWW:URL:<http://www.hometoys.com/emagazine/1997/08/introduction-to-the-cebus-communications-protocol/581>.
- [22] **Padrão CEBus** - [Em linha] [Consult. 4 set. 2014]. Disponível em WWW:URL:<http://www.cricte2004.eletrica.ufpr.br/edu/anterior/cd00/trab/cebus/>.
- [23] ECHELON CORPORATION - Introduction to the LonWorks® Platform. [s.d.]) 1–98.
- [24] TOSHIBA CORPORATION - Neuron Chip Local Operating Network LSIs. [s.d.]).
- [25] ECHELON CORPORATION - Common LonWorks Channel Types. [s.d.]).
- [26] POWERLINE CONTROL SYSTEMS - UPB Technology Description v1.4. [s.d.]).
- [27] **Powerline Control Systems Signs Seventh UPB Licensee Smart Home Ltd. - Powerline Lighting Systems - PCS** - [Em linha] [Consult. 9 set. 2014]. Disponível em WWW:URL:<http://www.pcslighting.com/news/press-releases/121-powerline-control-systems-signs-seventh-upb-licensee-smart-home-ltd>.
- [28] FERREIRA, A. M. A. - **Desenvolvimento de Infra-Estrutura de Comando Multifunções EIB-KNX Para Smartphone**. [S.l.] : Escola Superior de Tecnologia e Gestão Instituto Politécnico de Bragança, 2012
- [29] JOSUÉ, J. G. - **Projecto e Construção de um Sistema de Monitorização de Energia Eléctrica para uma Habitação**. [S.l.] : Universidade Nova de Lisboa Faculdade de Ciências e Tecnologia, 2010
- [30] ZAMORA-IZQUIERDO, M. A.; SANTA, J.; GÓMEZ-SKARMETA, A. F. - An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. 2010) 0–12.
- [31] JUING-HUEI SU, CHYI-SHYONG LEE, W.-C. W. - The Design and Implementation of a Low-cost and Programmable Home Automation Module. 2006) 0–6.
- [32] **Embedded Automation, Inc. - Products - mControl v3** - [Em linha] [Consult. 9 set. 2014]. Disponível em WWW:URL:<http://www.embeddedautomation.com/products/mcontrol.asp>.
- [33] **Home Control Software - HomeSeer Home Automation Systems** - [Em linha] [Consult. 9 set. 2014]. Disponível em WWW:URL:<http://www.homeseer.com/home-control-software.html>.
- [34] **Best Home Automation System | ASecureLife.com** - [Em linha] [Consult. 9 set. 2014]. Disponível em WWW:URL:<http://www.asecurelife.com/best-home-automation-system/>.
- [35] **Mordomus - Intelligent House Management** - [Em linha] [Consult. 9 set. 2014]. Disponível em WWW:URL:<http://www.mordomus.com/>.
- [36] SANTOS, J. P. - PROTOCOLO DE COMUNICAÇÃO SÉRIE EIA485. In **Informática Industrial 2010/2011**. p. 31–37.

- [37] NOVUS PRODUTOS ELETRÔNICOS - Problemas com a rede RS485? [s.d.].
- [38] MEDIDORES, K. - Conceitos Básicos de RS-485 e RS-422. [s.d.].
- [39] SANTOS, J. P. - PROTOCOLO DE COMUNICAÇÃO MODBUS. In **Informática Industrial 2010/2011**. p. 41–47.
- [40] MODBUS ORGANIZATION - **MODBUS Application Protocol v1.1b**. 1–51 p.
- [41] ELECTRONICS AND RADIO TODAY - **The Radio Spectrum** [Em linha] [Consult. 13 out. 2014]. Disponível em WWW:URL:[http://www.electronics-radio.com/articles/radio/basic\\_radio/radio\\_spectrum/radio\\_spectrum.php](http://www.electronics-radio.com/articles/radio/basic_radio/radio_spectrum/radio_spectrum.php).
- [42] BBC - The Spectrum and Its Uses. September (2006).
- [43] **Movicon** - [Em linha] [Consult. 13 out. 2014]. Disponível em WWW:URL:<http://www.mxcad.com/movicon>.
- [44] PROGEA - **Movicon BA supervision system for building automation** [Em linha], atual. 2013. a. [Consult. 13 out. 2014]. Disponível em WWW:URL:<http://www.progea.com/en-us/products/moviconbuildingautomation.aspx>.
- [45] PROGEA - **Pro.Energy** [Em linha] [Consult. 5 dez. 2014]. Disponível em WWW:URL:<http://www.progea.com/en-us/products/proenergy.aspx>.
- [46] PROGEA - **Movicon BA Integrated I/O Drivers** [Em linha], atual. 2013. b. [Consult. 13 out. 2014]. Disponível em WWW:URL:<http://www.progea.com/en-us/products/moviconbuildingautomation/integratediodrivers.aspx>.
- [47] Microchip Technology. (2011). *PIC18F26J50 Data Sheet*.
- [48] OLIMEX - **MOD-RS485 development board Users Manual**
- [49] MICROCHIP TECHNOLOGY - **MRF24J40** [Em linha] [Consult. 30 out. 2014]. Disponível em WWW:URL:<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en027752>.
- [50] NORDIC SEMICONDUCTOR - **nRF24L01+ Product Specification v1.0**
- [51] **WM22DIN.jpg (796 × 553 pixels)** - [Em linha] [Consult. 10 nov. 2014]. Disponível em WWW:URL:[http://www.shmcomms.co.uk/products/electricity\\_meters.htm#accordion](http://www.shmcomms.co.uk/products/electricity_meters.htm#accordion)



## Anexo 1 - Orçamento das licenças Movicon

### Licenças *runtime* SCADA

Referência	Descrição	PVP
RT11-64-B	Runtime Scada LITE 64 I/O bytes	680.00 €
RT11-64-F	Runtime Scada PRO <sup>12</sup> 64 I/O bytes	945.00 €

### Opção das licenças *runtime* SCADA

Referência	Descrição	PVP
OP-DL	Data Logger (for OEM Runtime, LT, BA) <sup>1</sup>	400.00 €
OP-MD	Multidriver (for Runtime LT, BA)	400.00 €
OP-DRV	Special Driver Activation (see list) <sup>2</sup>	400.00 €
OP-NET	Networking Option (for Runtime LT, BA)	400.00 €
OP-ALD	Alarm Dispatcher Option (SMS, Voice, Fax) <sup>3</sup>	400.00 €
OP-ALS	Alarm Statistics Analysis Option	500.00 €
OP-OXC	OPC DA and XML DA Server function	500.00 €
OP-UAW	Function OPC UA (Data Access) Client <sup>4</sup>	500.00 €
OP-RED	Redundancy function <sup>4</sup>	700.00 €
OP-DEV	Editor function <sup>5</sup>	700.00 €

1. A opção inclui o *designer* de relatórios.

2. Em adição ao pacote de drivers I/O incluídas gratuitamente no Movicon, a versão 11.4 fornece um pacote de drivers especiais que requerem, para o funcionamento, a ativação da licença de *runtime*. A lista de drivers pode ser consultada em <http://www.progea.com/en-us/products/iodriverslist.aspx>.

3. A opção *Alarm Dispatcher* inclui a opção *Text-to-Speech*, por defeito, em inglês. Esta opção inclui fonema em inglês. A Progea pode incluir fonemas alternativos para italiano, alemão, com o mesmo preço, se especificado quando efetuada a encomenda.

4. A funcionalidade cliente OPC UA (especificação Data Access) é baseada em *software* de terceiros.

5. A função de redundância requer uma licença de servidor e fornece a configuração com 2 licenças de servidor (Servidor secundário e Servidor Primário) com as duas opções ativas. As opções não estão disponíveis para licenças de clientes.

**Nota:** Dependendo do projeto, a estes preços, podem ser considerados descontos até 30%.

<sup>12</sup> Se o projeto usar *Data Logger*, *Networking* ou funções *MultiDriver* será necessário uma licença PRO.

Licenças *runtime* Windows CE<sup>13</sup>

Referência	Descrição	PVP
RT11-CE-B	CE Runtime Lite (1024 bytes)	350.00 €
RT11-CE	CE Runtime (4096 bytes)	520.00 €

Nas licenças estão incluídos:

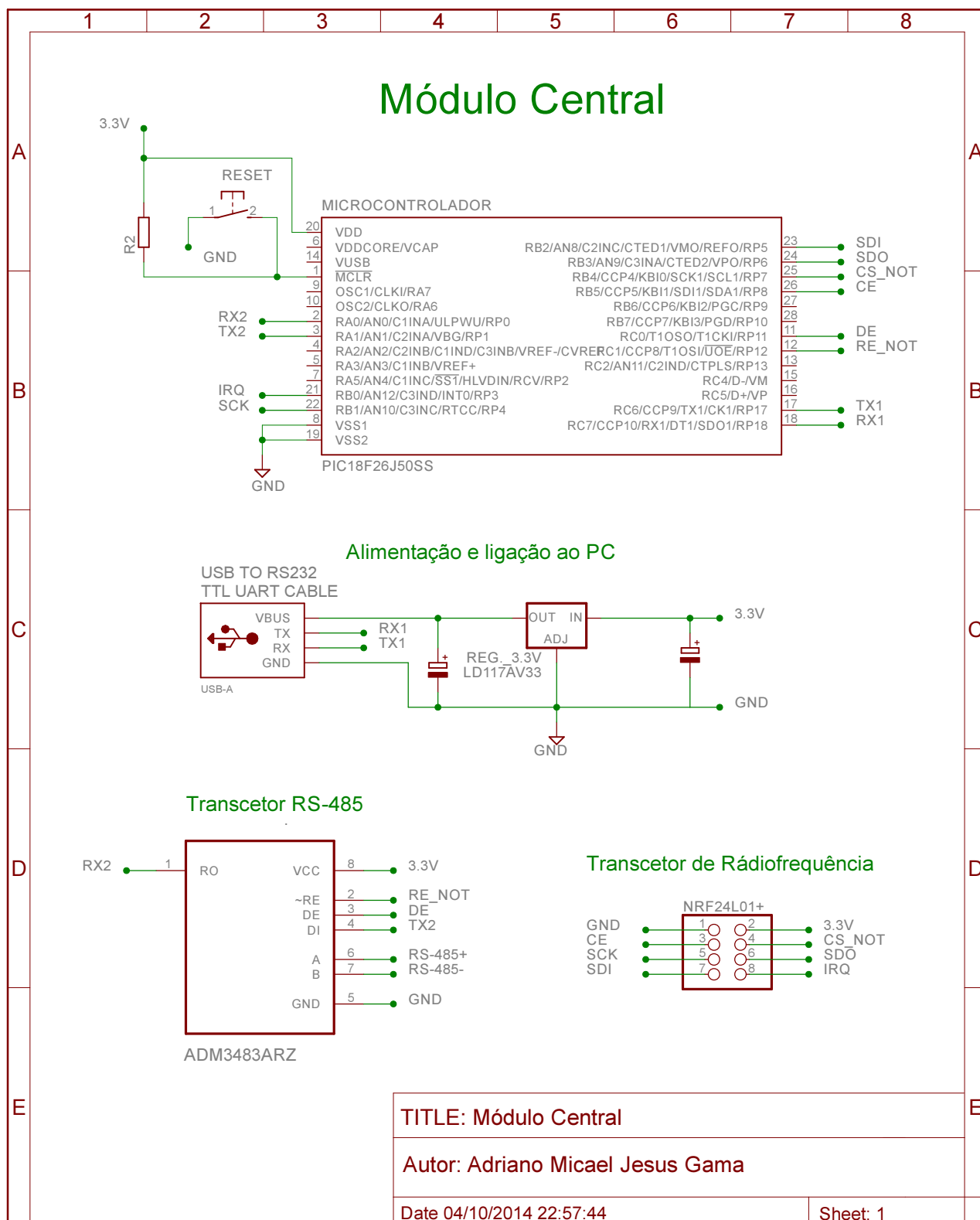
1. O mesmo projeto XML corre num PC *desktop*.
2. Depuração remota.
3. *Download/Upload* no dispositivo via TCP, FTP ou *Active Sync*.
4. Gráficos vetor ou BPM.
5. Gestão de alarmes.
6. Registo histórico (Texto, XML ou ADOCE servidor SQL).
7. Notificação de alarmes (SMS e E-mail).
8. Modelo cliente/servidor.
9. Cliente OPC.
10. Gestão de instruções.
11. Bases de dados (Texto, XML ou ADOCE servidor SQL).
12. Relatórios textuais.
13. Análise de dados e de tendências.
14. Linguagem VBA.
15. Linguagem lógica (AWL - Step® *Instruction List*).
16. Objetos de calendarização.
17. Objetos de eventos.
18. Objetos de visualização de câmaras IP.
19. Mudança dinâmica de linguagem.
20. Gestão de utilizadores e de *passwords* (1024 níveis, 16 áreas) com CFR21.
21. Teclado virtual táctil padrão e personalizável.
22. Uso máximo de 2 *drivers* em simultâneo (4 no total).
23. Clientes WEB (máximo de 2).
24. Máximo de 4096 bytes de I/O.

---

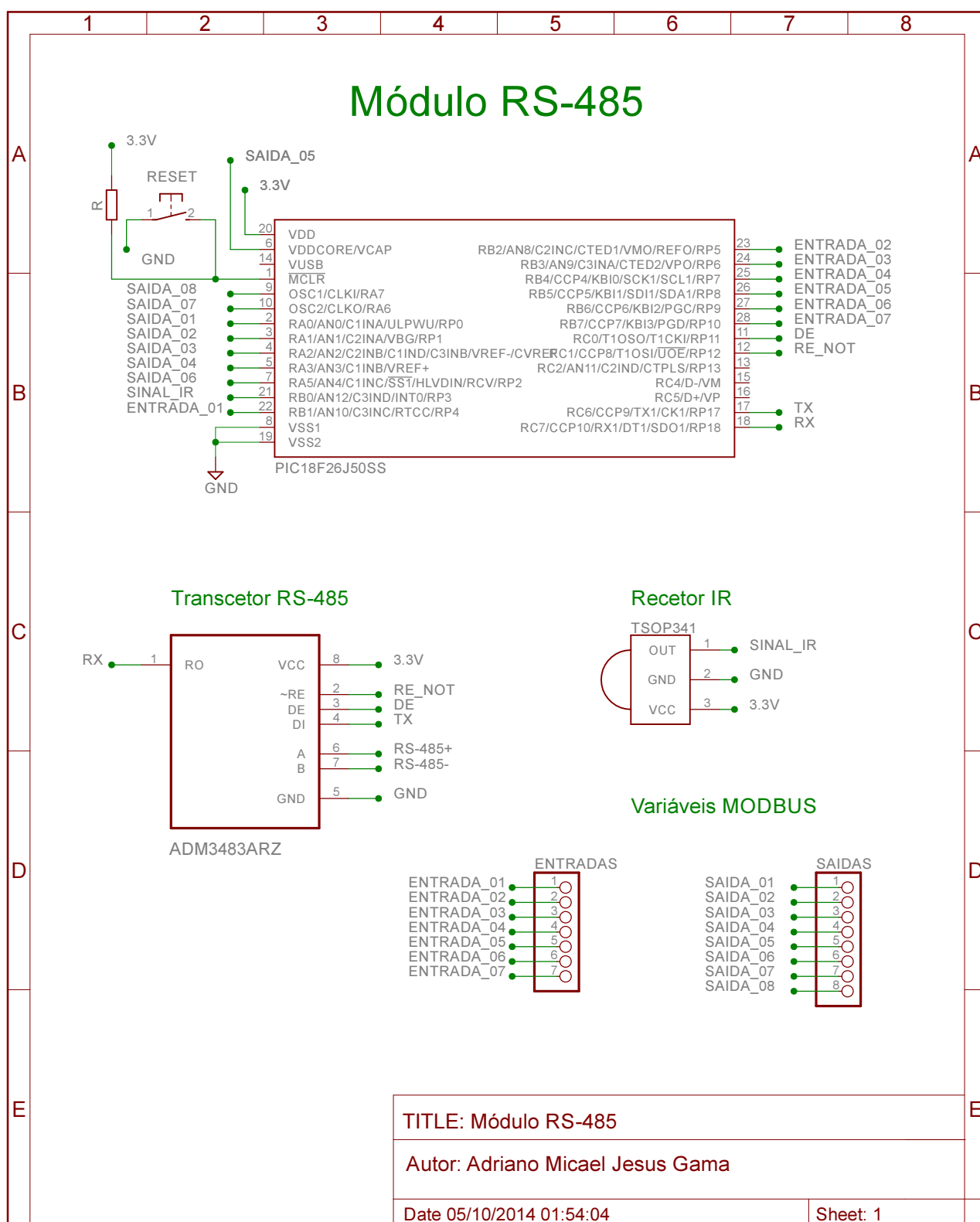
<sup>13</sup> Licenças para Windows CE 5.0, 6.0, 7.0 ou Windows Mobile (PocketPC, PDA, Smartphone).



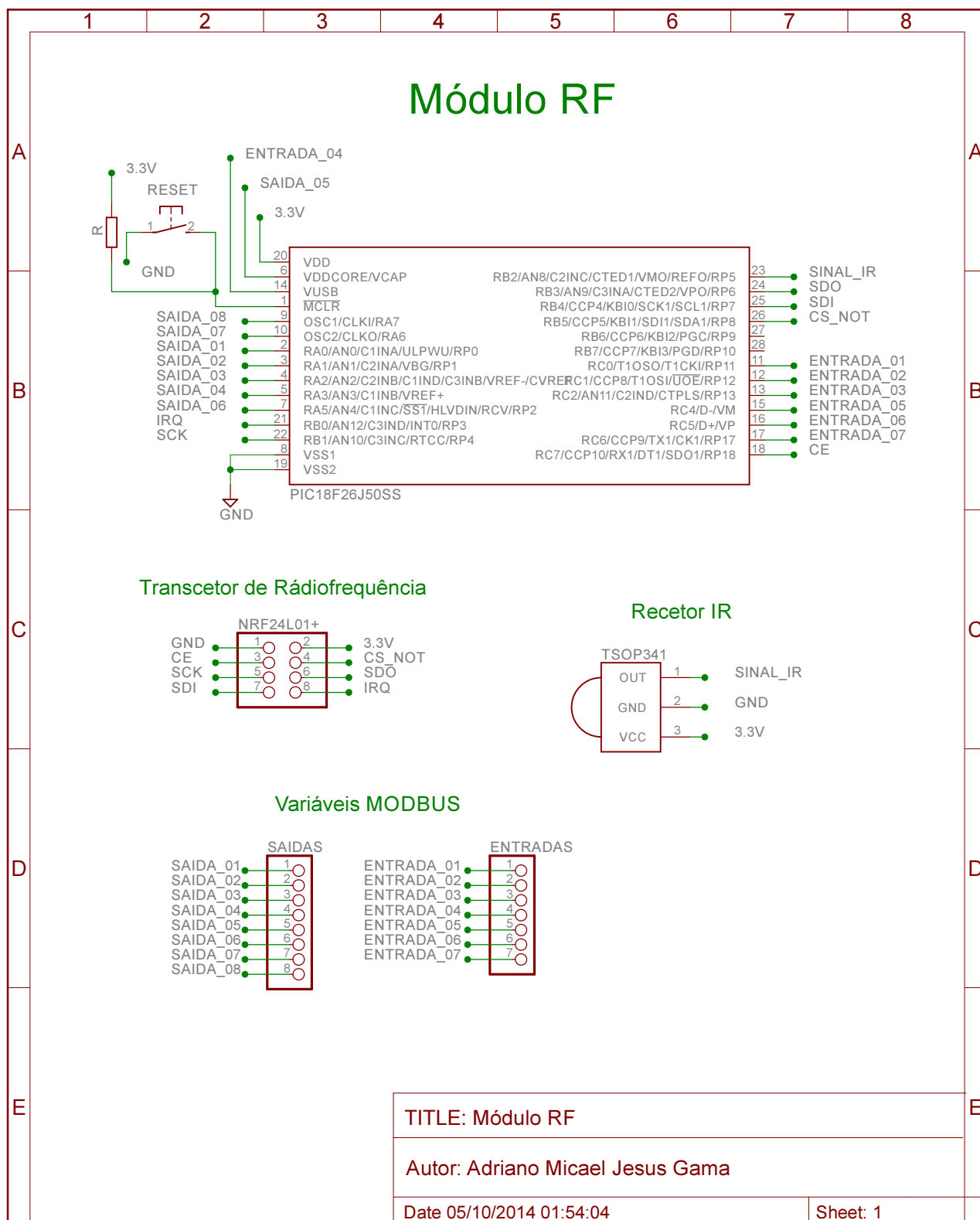
## Anexo 2 - Esquema elétrico do módulo central



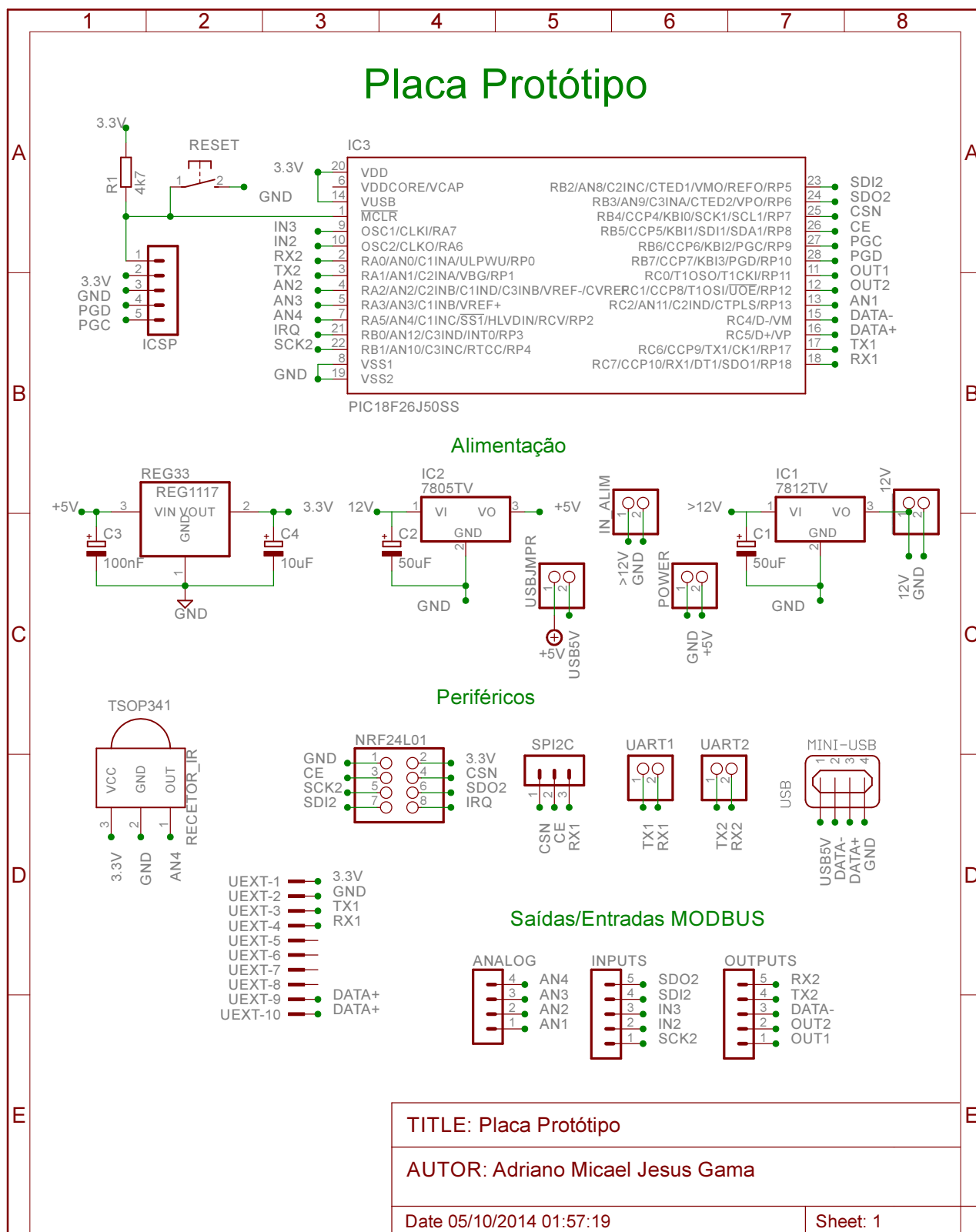
## Anexo 3 - Esquema elétrico do módulo RS-485



## Anexo 4 - Esquema elétrico do módulo RF



## Anexo 5 - Esquema elétrico da placa protótipo



## Anexo 6 - Orçamento da placa protótipo



SEED STUDIO

### Quotation

Quote Number : Quote 092614

Quote date: 2014/9/26

**From: Seeed Technology Inc.**

**To: Adriano Gama**

**Address:** 5th Floor, 8th Building, Shiling industrial Park  
XiLi Town, NanShan dist.  
Shenzhen China

**Address:**

**Contacts:** Mandy Xiang

**Contacts:** Adriano Gama

**Tel:** +86 755 29338658

**Tel:**

**Email:** mandy.xiang@seeedstudio.com

**Email:** amjg@ua.pt

**Post Code:** 518055

**Post Code:**

Nu.	Item	Description	Unit Cost (USD\$)	Remark
1	Placa Protótipo	PCBA	US\$11.50	20PCS
2	Placa Protótipo	PCBA	US\$7.00	100PCS
3	Placa Protótipo	PCBA	US\$6.40	500PCS
4		Shipping address: Universidade de Aveiro Campus Universitário de Santiago 3810-193 Aveiro	US\$35.00	

#### Notes;

- Tax & Shipping Cost: the quotation is not included any customs fee,taxes expense and shipping cost;
- Valid Period: keep the quotation valid period three months,from 2014/9/26 to 2014/12/26
- The unit price is based on latest information what you provided, and is subject to change according to the finalized product specification;